

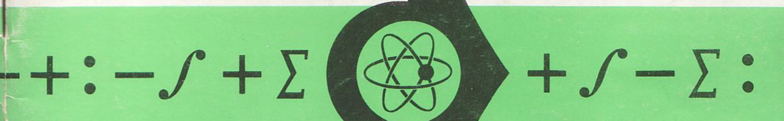

VEB
KOMBINAT ZENTRONIK
RECHENELEKTRONIK
MEININGEN/ZELLA MEHLIS

Görtz

# PROGRAMMIERUNG UND BEDIENUNG DES DIGITALEN KLEINRECHNERS C 8205

## HEFT 7

## FESTKOMMA-INTERPRETIERENDES PROGRAMMSYSTEM MIT VARIABLER SPEICHER-WORTLÄNGE FIPS v



# Programmierung und Bedienung

# des digitalen Kleinrechners C 8205

## Heft 7

## Festkommainterpretierendes Programmsystem

## mit variabler Speicherwortlänge

## FIPS v

# Inhaltsverzeichnis

# 0. Einführung

Ein Programm für Rechenautomaten besteht aus einer Folge von elementaren Anweisungen (Befehlen), die nacheinander vom Automaten auszuführen sind. Durch Sprungbefehle kann die fortlaufende Abarbeitung unterbrochen und an einer gewünschten Stelle fortgesetzt werden, insbesondere kann so ein Programmteil mehrfach durchlaufen werden (zyklische Programme).

Die Liste der zulässigen Befehle ist durch den Automatentyp vorgegeben. Sie beeinflußt entscheidend den erforderlichen Programmieraufwand und den zur Abspeicherung des Programmes benötigten Speicherraum.

In der Befehlsliste des C 8205 sind Befehle wie Eingabe und Ausgabe eines Dezimalwortes, Multiplikation und Division nicht enthalten. Für diese Operationen werden deshalb Teilprogramme verwendet, die von beliebigen Stellen aufgerufen werden können und nach Abarbeitung an die aufrufende Stelle zurückführen (Unterprogramme). Da diesen Unterprogrammen gewöhnlich bestimmte Parameter übermittelt werden müssen, ergeben sich relativ lange Befehlsfolgen, und Programmierungsfehler treten leicht auf.

Werden hingegen erst bei Abarbeitung des Programmes Pseudobefehle durch ein Entschlüsselungsprogramm analysiert und damit die Abarbeitung entsprechender Maschinenbefehlsfolgen gesteuert (interpretierende Abarbeitung), so kann auf dem C 8205 ein leistungsfähigeres Befehlssystem simuliert werden. Das Entschlüsselungsprogramm selbst erfordert nur wenig Speicherplatz, und durch die Programmierung mit Pseudobefehlen wird die Länge der Objektprogramme wesentlich reduziert. Zu beachten ist jedoch, daß bei Abarbeitung jedes Befehles eine zusätzliche Entschlüsselungszeit erforderlich ist.

Ein System, bestehend aus dem Entschlüsselungsprogramm, den erforderlichen Maschinenbefehlsfolgen analog den Unterprogrammen bei direkter Abarbeitung und der Eingabeorganisation zur Programmeingabe, stellt also für den Anwender in Verbindung mit dem C 8205 einen Rechenautomaten mit neuen Parametern dar. Ein Systemwechsel ist in wenigen Minuten möglich und erlaubt eine optimale Anpassung an die vorliegenden Probleme.

Ein solches System stellt das FIPS v (festkommainterpretierendes Programmsystem mit variabler Speicherwortlänge) dar. Sein Einsatzgebiet wird durch folgende Eigenschaften gekennzeichnet:

- leistungsstarke Einadreßbefehle mit weitgehend analytischem Aufbau
- 2 Rechenregister, 8 Indexregister, 8 Druckregister
- günstige Möglichkeiten der Befehlsmodifikation
- Festkommaverarbeitung bis zu 18stelliger Dezimalzahlen
- optimale Ausnutzung des Arbeitsspeichers durch programmierbare Wortlänge (bis 13 Dezimalstellen) bei Transporten von und zum Hauptspeicher
- Eingabe und Ausgabe von Dezimalzahlen auch im Satz
- bei Nichtbedarf anderweitige Nutzung des Speicherplatzes der Systemteile „Textausgabe" und „Division" sowie der Hilfsprogramme möglich.

Das System ist besonders für ökonomische und ökonometrische Berechnungen geeignet, wobei eine pfenniggenaue, den kaufmännischen Rundungsregeln entsprechende Verarbeitung auch großer Beträge gewährleistet ist.

Die Programmierung im System „FIPS v" erfordert keine Kenntnis der Maschinensprache des Automaten, sofern nicht Programmteile in dieser Sprache oder mit speziellen, hier nicht aufgeführten Sonderbefehlen eingefügt werden sollen.

„FIPS v" ist nur für die Eingabe im R 300-Code ausgelegt.

# 1. Informationsdarstellung

## 1.1. Externe Darstellung

Die Eingabe von Informationen am C 8205 (R 300-Variante) ist durch Tasten- und Schalterbetätigung am Bedienungsteil oder Schreibwerk und Eingabe von Lochbändern über die Lochbandleser möglich. Das Lesen eines Zeichens vom Lochband ist der Betätigung der durch den externen R 300-Code zugeordneten Taste der Befehlstastatur bzw. des Schreibwerkes gleichwertig (Tab. 1). Die Stellung der Selektoren „SP" und „BS" kann nur manuell verändert werden.

Die Darstellung von Informationen in einer für die Eingabe geeigneten Form sei hier externe Darstellung genannt.

### 1.1.1. Dezimalzahlen

Dezimalzahlen bestehen aus einer Folge von Zeichen der Menge
a) 0 1 2 3 4 5 6 7 8 9
b) –
c) , ZR GB KB SPE IZ
d) WR NS WS + Leserstopp
und werden durch eine Wortmarke (Punkt oder Tabulator) abgeschlossen.

Nach dem Zeichen „–" darf nur „–" (damit wird bei der Eingabe der bereits eingegebene Wortteil gelöscht) oder eine Wortmarke (negative Werte) folgen.

Die Zeichen der Gruppe c) haben keinerlei Einfluß auf den Wert der Zahl, die Zeichen der Gruppe d) besitzen die Wertigkeit „0" und haben somit keinen Einfluß auf den Wert der Zahl, wenn sie vor der ersten von Null verschiedenen Ziffer stehen.

Sätze von Dezimalzahlen werden durch eine Satzmarke Sm (WRZL) oder Blockmarke Bm abgeschlossen. Die Wirkung von „:" hängt von dem aufrufenden Befehl ab.

Durch „IS", „IB" oder „'" kann jeweils ein noch nicht abgeschlossener Satz gelöscht werden. „SP" und „LOA" führen zum Verlassen des Eingabeprogrammes und zur Verfälschung des Eingabewertes. Alle hier nicht aufgeführten Zeichen verfälschen den Eingabewert.

### 1.1.2. Oktalzahlen

In Oktalzahlen dürfen die Ziffern 8 und 9 nicht auftreten, es sind höchstens 11 Ziffern zugelassen. Im übrigen gelten alle Angaben zu Punkt 1.1.1.

Als Oktalzahlen können z. B. Programmkonstanten (Normalworte) eingegeben werden. Durch formale Interpretation des Inhaltes jeder Zelle als (absolute) Oktalzahl kann der Inhalt eines Speicherbereiches (z. B. Programm einschließlich aller Konstanten) zur Zwischenspeicherung auf Lochband ausgegeben und später wieder eingelesen werden. Da Maschinen- und Systembefehle triadenweise verschlüsselt werden, läßt sich deren Bedeutung aus der Oktaldarstellung ohne weiteres erkennen.

### 1.1.3. Befehlsdarstellung

#### 1.1.3.1. Adressierung

Die 4096 Speicherplätze des C 8205 werden nicht dezimal, sondern oktal numeriert. Für die oktale Adressierung sind keine weiteren Kenntnisse über das Oktalsystem erforderlich, wenn man nur beachtet, daß im Oktalsystem die Ziffern 8 und 9 nicht auftreten. So folgt

auf Adresse 7 sofort 10 und auf 5777 Adresse 6000. Damit ergibt sich eine durchgehende Numerierung von 0000–7777. Jeder Adresse a ist eindeutig eine Zelle des Hauptspeichers zugeordnet und umgekehrt. Man spricht von Zelle a oder auch nur von „a". Der Inhalt der Zelle a wird mit ⟨a⟩ bezeichnet. Die Inhalte sämtlicher Register werden ohne Inhalt-Klammern angegeben.

Für den Anwender von FIPS v stehen die Zellen 2300–7337 frei zur Verfügung. Auf den Plätzen 0000–2277 stehen die Maschinenbefehle, die zur Arbeit des Systems erforderlich sind, davon werden die Plätze 2040–2277 nur für die Systembefehle „Division" und „Textausgabe" benötigt. Kann auf diese Befehle verzichtet werden, können diese Plätze nach Einlesen eines Änderungsstreifens anderweitig genutzt werden. Die Plätze 7340–7537 enthalten Hilfsprogramme, die übersprochen werden können, wenn sie bei Abarbeitung getesteter Programme nicht benötigt werden.

Die Spuren 7540–7777 werden stets vom System benötigt und dürfen vom Programmierer nicht frei belegt werden. Die Rechenregister, Indexregister und Druckregister werden durch folgende Arbeitszellen simuliert:

| | |
|---|---|
| AC | 7607, 7610 und 7612 |
| R | 7611 und 7613 |
| I 0 | 7760 |
| . | . |
| . | . |
| . | . |
| I 7 | 7767 |
| D 0 | 7630 |
| . | . |
| . | . |
| . | . |
| D 7 | 7637 |

Indexregister und Druckregister sind beliebig adressierbar, der Transport eines Wortes von AC nach R ist als Normalwort mit Adresse 7611 möglich, wenn der Inhalt von 7613 eindeutig „0" ist.

Der Pseudobefehlszähler PZ (Zelle 7602) wird ebenso wie I 0 bei Erreichen des Großen Eingabeprogrammes gelöscht, deshalb wird PZ zusätzlich in Zelle 7600 abgespeichert.

Als Pseudobefehlsregister BR fungiert die Spur 7640–7677, unverschoben steht der aufgerufene Befehl in Zelle 7645. Während der Abarbeitung wird BR bei einigen Befehlen verändert.

### 1.1.3.2. Maschinenbefehle

Maschinenbefehle bestehen aus einer 4stelligen oktalen Adresse und 3 Triaden für Operationstyp, Operation und Resultatbehandlung bzw. Löschanweisung. Näheres siehe „Anleitung zur Programmierung in der Maschinensprache".

### 1.1.3.3. Systembefehle

Systembefehle bestehen allgemein aus 11 Triaden, davon 4 Triaden zur Darstellung der 4stelligen oktalen Adresse. Die Stellung der Adresse ist bei Maschinen- und Systembefehlen identisch. Die formale Übereinstimmung von Oktalzahlen, Maschinen- und Systembefehlen ermöglicht eine einheitliche Ein- und Ausgabe dieser Worte.

Eine übersichtliche Schreibweise wird durch Abgrenzung der Adresse mit „ZR" (hat bei Eingabe keine Wirkung) erreicht.

### 1.1.4. Textdarstellung

Zur rationellen Speicherung von alphanumerischem Text können 5 Zeichen in einem Oktalwort codiert werden. Dabei ist jedem Zeichen eine zweistellige Schlüsselnummer entsprechend Tab. 2 zuzuordnen. Diese Schlüsselnummern werden zu je fünf von links nach rechts aneinandergefügt. Die Schlüsselnummer 77 markiert das Ende des Textes. Weniger als 5 Schlüsselnummern bei Texten sind durch beliebige Ziffern zu ergänzen. Die 11. Oktalstelle ($t_{10}$) kann bei Bedarf beliebig belegt werden; sie ist im allgemeinen gleich 0.

Ein so verschlüsselter Text wird nach Eingabe in den Speicher durch einen entsprechenden Systembefehl mit Angabe der Anfangsadresse aufgerufen. Eine maschinelle Verschlüsselung ist über ein gesondertes Programm möglich.

## 1.2. Interne Darstellung

Der einheitliche Speicher des C 8205 ist in 128 Spuren zu je 32 Sektoren aufgeteilt. Die kleinste adressierbare Informationseinheit ist das Wort. Es stellt den Inhalt einer Zelle, in Ausnahmefällen den Inhalt eines Teiles einer Zelle (Füllwort, Miniwort) oder zweier Zellen (Doppelwort), dar. Jede Information wird intern rein binär dargestellt. Ihre Interpretation ergibt sich erst aus ihrer Verarbeitung.

### 1.2.1. Triaden

Der Inhalt jeder Zelle besteht, unabhängig von seiner Interpretation, aus 33 bit, die dezimal mit 0–32 von links nach rechts durchnumeriert werden. Jede dieser 33 Dualstellen kann mit den Ziffern 0 oder 1 besetzt sein, so daß es insgesamt $2^{33} = 8{,}589934592 \times 10^{9}$ verschiedene Maschinenworte gibt. Zweckmäßig faßt man bei der Darstellung je 3 bit zu einer Triade zusammen. Abb. 1 zeigt Bezeichnung und Zuordnung der Wortstellen sowie ein Maschinenwort in Bit- und Triadendarstellung.

| Triaden | $t_{10}$ | | | $t_9$ | | | $t_8$ | | | $t_7$ | | | $t_6$ | | | $t_5$ | | | $t_4$ | | | $t_3$ | | | $t_2$ | | | $t_1$ | | | $t_0$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bits | $Z_0$ | $Z_1$ | $Z_2$ | $Z_3$ | $Z_4$ | $Z_5$ | $Z_6$ | $Z_7$ | $Z_8$ | $Z_9$ | $Z_{10}$ | $Z_{11}$ | $Z_{12}$ | $Z_{13}$ | $Z_{14}$ | $Z_{15}$ | $Z_{16}$ | $Z_{17}$ | $Z_{18}$ | $Z_{19}$ | $Z_{20}$ | $Z_{21}$ | $Z_{22}$ | $Z_{23}$ | $Z_{24}$ | $Z_{25}$ | $Z_{26}$ | $Z_{27}$ | $Z_{28}$ | $Z_{29}$ | $Z_{30}$ | $Z_{31}$ | $Z_{32}$ |
| Maschinenwort | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| | 3 | | | 5 | | | 7 | | | 0 | | | 4 | | | 1 | | | 6 | | | 2 | | | 3 | | | 7 | | | 2 | | |

Abb. 1

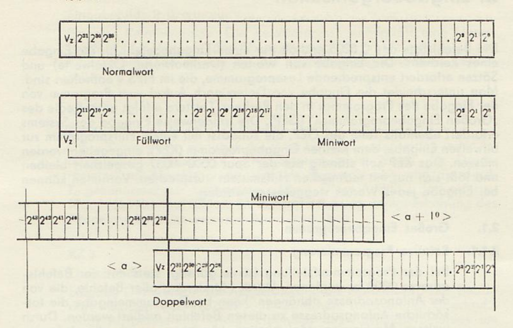
### 1.2.2. Dezimalzahlen

Durch das Eingabeprogramm wird die externe Darstellung von Dezimalzahlen in die interne Dualdarstellung konvertiert.

Nach „Eingabe Dezimalwort" steht der Betrag dieser Dualzahl in den Zellen 7610 (32 bit) und 7612 (höherwertige bit), ein negatives Vorzeichen in der Stelle $z_0$ der Zelle 7607. Die Zellen 7607, 7610 und 7612 bilden zusammen den Pseudoakkumulator AC. Von AC können Dezimalzahlen in das Rechenregister R (Zellen 7611 und 7613, Vorzeichen in $z_0$ von 7611) und in die Hauptspeicherzellen transportiert werden.

Im Hauptspeicher kann in einer Zelle ein Normalwort (Betrag in $z_{32}$–$z_1$, Vorzeichen in $z_0$) oder ein Miniwort (Betrag in $z_2$–$z_{13}$, Vorzeichen in $z_0$) und ein Füllwort (in $z_{12}$–$z_1$, nur positiv) stehen. Ein Doppelwort besteht aus einem Normalwort und dem Füllwort der darauffolgenden Zelle, letzteres mit dem Faktor $2^{32}$ bewertet. Die freien Stellen dieser Zelle können durch ein zusätzliches Miniwort belegt werden (Abb. 2).

Abb. 2




Da ein Komma bei der Eingabe nicht ausgewertet und auch nicht abgespeichert wird, muß der Programmierer die Stellung des Kommas kennen und bei der Programmierung berücksichtigen. So können nur Zahlen mit gleicher Kommastellung ohne vorherige Verschiebungen addiert oder subtrahiert werden, bei der Multiplikation bzw. Division ist der gewünschte Kommawert des Ergebnisses in Abhängigkeit von den Kommawerten der Operanden durch eine Verschiebung des Produktes bzw. des Dividenden vor der Division zu realisieren. Bei der Ausgabe ist die Stellung des Kommas im Druckregister anzugeben.

Die im System „FIPS v" angewandte interne Darstellung von Dezimalworten unterscheidet sich auch beim Normalwort durch die Betragsdarstellung von der Interpretation durch Maschinenbefehle, so daß eine Addition bzw. Subtraktion von Normalworten durch die entsprechenden Maschinenbefehle nicht möglich ist, wenn negative Werte auftreten.

# 2. Eingabeorganisation

Die Befehlsliste des C 8205 enthält nur einen Eingabebefehl für die Eingabe eines Zeichens. Die Eingabe von Worten (Dezimalworte, Oktalworte) und Sätzen erfordert entsprechende Leseprogramme, die im FIPS v enthalten sind. Man unterscheidet die Eingabe von Daten nach Aufruf vom Programm von der Eingabe des Programmes in den Rechner. Letztere erfolgt unter Regie des „Großen Eingabeprogrammes" (GEP). Die triadenweise Eingabe des Systems erfordert ebenfalls Teile des GEP, die zunächst mit einem Hilfsprogramm zur bitweisen Eingabe, dem Kleinen Eingabeprogramm (KEP), eingegeben werden müssen. Das KEP soll ständig auf der Spur 0000–0037 gespeichert bleiben und läßt sich nur mit technischen Hilfsmitteln aufsprechen. Vornullen können bei Eingabe jedes Wortes weggelassen werden.

## 2.1. Großes Eingabeprogramm

### 2.1.1. Relative Programmierung

Bei der Programmaufstellung ist es günstig, stets mit der Befehlsadresse 0000 zu beginnen. Durch Markierung aller Befehle, die von der Anfangsadresse abhängen, kann bei Programmeingabe die tatsächliche Anfangsadresse zu diesen Befehlen addiert werden. Durch weitere Markierungsmöglichkeiten können so auch die Anfangsadressen von Konstanten- und Datenfeldern und von benötigten Unterprogrammen zunächst relativ mit 0000 programmiert werden. Erst nach Kenntnis der Länge von Programm und Feldern erfolgt die Zuordnung der absoluten Speicherbereiche.

Die Markierung sämtlicher Eingabeworte in Triadenform bei Eingabe mit dem Großen Eingabeprogramm erfolgt durch die Angabe eines zweiten Kennzeichens, das die Nummer der zugehörigen Leitadresse, die addiert werden soll, angibt.

Die 0. Leitadresse ist stets gleich „0", alle übrigen Leitadressen müssen vor Eingabe der entsprechend markierten Triadenworte dem Rechner mitgeteilt werden. Ihre Abspeicherung erfolgt in den zugeordneten Indexregistern.

### 2.1.2. Eingabekennzeichen

Zur Unterscheidung der unterschiedlichen Informationsdarstellungen (Oktalzahlen, Dezimalworte und -sätze) beginnt jede Eingabe im GEP mit einem 1. Kennzeichen, dem teilweise ein 2. Kennzeichen folgt.

Eingabezeichen dienen nur zur Steuerung der Eingabe und werden nicht mit abgespeichert.

**KZ 21**

Eingabebild: 2 1 $b_1b_2b_3b_4$ $a_1a_2a_3a_4$.

Eingabe 1. Leitadresse. Die auf die Adressenstellen folgenden 3 Oktalziffern werden vom Eingabeprogramm mit Nullen aufgefüllt. Außerdem beginnt ab der durch $a_1a_2a_3a_4$ symbolisierten Adresse die Abspeicherung der darauffolgenden Worte mit 1. KZ = 0, 3, 4 oder 8.

**KZ 22–27**

Eingabebild: 2 $K_2$ $b_1b_2b_3b_4$ $a_1a_2a_3a_4$.

Eingabe 2.–7. Leitadresse. Die auf die Adressenstellen des Oktalwortes folgenden 3 Ziffern werden vom Eingabeprogramm mit Nullen belegt.

Die Eingabe dieser Leitadressen hat keinen Einfluß auf die Abspeicheradresse.

**KZ 20**

Eingabebild: 2 0 $a_1a_2a_3a_4$.

Eingabe Relativadresse. Die Relativadresse ermöglicht eine Unterbrechung der fortlaufenden Abspeicherung. Sie gibt relativ zur 1. Leitadresse die Zelle an, bei der die Abspeicherung fortgesetzt werden soll.

**KZ 0$K_2$**

Eingabebild: 0 $K_2$ $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ $b_5b_6b_7$.

Diese Kennzeichen erlauben die Eingabe von Triadenworten jeder Art (Oktalzahlen, Maschinen- und Systembefehle, Inhalt der Druckregister) zur Abspeicherung. Vor der Abspeicherung wird die durch $K_2$ gekennzeichnete Leitadresse addiert. Diese Addition darf nicht zu Maschinenüberlauf führen. Nach Abspeicherung erfolgt Erhöhung des Abspeicherbefehles um eine Adresseneinheit.

**KZ 3**

Eingabebild: 3 $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$.

Eingabe Dezimalwort zur Abspeicherung als Normalwort. Der Betrag darf höchstens $4{,}294967289 \times 10^9$ erreichen, sonst erfolgt Überlauf. Nach Abspeicherung erfolgt Erhöhung des Abspeicherbefehles um eine Adresseneinheit.

**KZ 4$K_2$**

Eingabebild: 4 $K_2$ $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ $b_5b_6b_7$.
$b_1b_2b_3b_4$ $a_1a_2a_3a_4$ $b_5b_6b_7$.
.
.
.
$b_1b_2b_3b_4$ $a_1a_2a_3a_4$ $b_5b_6b_7$. :

Eingabe von Triadenworten im Satz. Das Kennzeichen $K_2$ wirkt auf alle Worte gleichmäßig. Jedes Wort wird nach Eingabe abgespeichert und der Abspeicherbefehl jeweils um eine Adresseneinheit erhöht. Als Endmarke kann außer : auch Bm oder Sm gegeben werden, bei Sm wird jedoch noch ein Wort „0–." zusätzlich abgespeichert.

**KZ 8**

Eingabebild: 8 $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$.
$x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$.
$x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$. :

Eingabe von Dezimalzahlen im Satz. Jedes Wort wird nach Eingabe abgespeichert und der Abspeicherbefehl jeweils um eine Adresseneinheit erhöht. Als Endmarke kann auch Bm oder Sm gegeben werden, bei Sm wird jedoch noch ein Wort „0–." zusätzlich abgespeichert.

**KZ 1$K_2$**

Eingabebild: 1 $K_2$ $a_1a_2a_3a_4$ $b_5b_6b_7$.

Maschinen-Sofortbefehl. Das eingelesene Triadenwort wird nach Addition der Leitadresse nicht abgespeichert, sondern als Maschinenbefehl interpretiert, der sofort ausgeführt wird. Handelt es sich nicht um einen Sprung mit erfüllter Bedingung, wird anschließend die Eingabe eines 1. KZ erwartet. Erlaubt insbesondere den Ansprung eines Maschinenprogrammes.

**KZ 5$K_2$**

Eingabebild: 5 $K_2$ $b_1b_2b_3b_4$ $a_1a_2a_3a_4$ $b_5b_6b_7$.

System-Sofortbefehl. Das eingelesene Triadenwort wird nach Addition

der Leitadresse nicht abgespeichert, sondern als Systembefehl interpretiert, der sofort ausgeführt wird.

Wird eine Systembefehlsfolge nur einmal durchlaufen, braucht sie somit nicht abgespeichert zu werden, sondern kann direkt vom Lochband abgearbeitet werden. Eine spezielle Anwendung stellt die Eingabe von Miniworten, Füllworten und Doppelworten als Programmkonstanten dar. Um Störungen bei der Arbeit mit dem Vergleichsprogramm zu vermeiden, wird die Anordnung am Ende des Programmlochstreifens und Angabe des Eingabekanales 1 empfohlen. Bei Angabe des Kanales 2 bzw. 3 (kein Anwahlwechsel) wird der Befehl einschließlich Abspeicherung auch bei Vergleichsprogramm ausgeführt, sonst erfolgt bei gesperrtem Leser 1 „Halt". Der System-Sofortbefehl „Eingabe Wort" zerstört den Abspeicherbefehl des GEP.

**KZ 6 oder 7 oder 9**

führen zur Überlaufanzeige.

**KZ „."**

Es wird ein Maschinenwort „0" abgespeichert und der Abspeicherbefehl um eine Adresseneinheit erhöht.

**KZ „,"**

Es erfolgt keine Abspeicherung, sondern lediglich eine Erhöhung des Abspeicherbefehles um eine Adresseneinheit.

**KZ „:"**

Das gespeicherte Systemprogramm wird, beginnend mit der in I 7 gespeicherten Adresse, abgearbeitet. Damit kann z. B. von externer Programmbearbeitung zur internen Arbeit übergegangen werden.

**KZ B0**

Eingabebild: B 0 $a_1a_2a_3a_4$. $w_1w_2w_3w_4$.

Beginn der interpretierenden Abarbeitung eines Programmes bei Zelle a. Vor Ausführung des in Zelle w gespeicherten Systembefehles erfolgt Ausdruck von PZ und AC (letzterer dezimal). Nach erneuter Eingabe einer Wartezelle w (über Tastatur) wird die Programmabarbeitung fortgesetzt. Wird kein derartiger Ausdruck gewünscht, braucht als zweite Adresse nur „." gegeben werden, da von Zelle 0000 niemals ein Systembefehl entnommen wird.

Für den Ausdruck ist die Speicherung der Hilfsprogramme mindestens ab Platz 7500 erforderlich.

**KZ B1**

Nach Eingabe von B1 addiert der Automat formal alle Plätze des Systems (ohne Hilfsprogramme) mit dem Sonderbefehl „Addition Spur", die Summe muß stets gleich Null sein. Ist das System richtig gespeichert, wartet der Automat nach einigen Sekunden Rechenzeit auf die Eingabe eines neuen 1. KZ, sonst erfolgt Stopp. Nach „Start" Eingabe 1. KZ. Die Summenbildung erfolgt in I1.

**1. KZ D**

Mit diesem Kennzeichen können einige Hilfsprogramme aufgerufen werden, wenn diese abgespeichert sind.

D0 Ausdruck R dezimal
D1 Ausdruck Inhalt der Zelle 7777 oktal
D2 Ausdruck PZ und AC
D3 Bereichsausschreiben oktal

D4 Bereichsausschreiben dezimal
D5 Vergleichsprogramm / oktaler Ausdruck
D6 Vergleichsprogramm / dezimaler Ausdruck
D8 Bereichsausstanzen oktal

Einzelheiten zu den Hilfsprogrammen siehe später.

**KZ e**

Lochbandende, führt zur Eingabe 1. KZ über die Tastatur.

**KZ /**

Lochband-Stopp. Ist der Selektor „BS" eingerastet, wird die Eingabe bei Erkennen dieses Zeichens unterbrochen. Fortsetzung der Eingabe mit 1. KZ nach „Start" oder Lösen des Selektors.

**KZ c**

Nach diesem Kennzeichen ist das Einlesen eines beliebigen Kommentars, bestehend aus Buchstaben, Ziffern, Wortmarken und den in Punkt 1.1.1. c) und d) aufgeführten Zeichen völlig wirkungslos. Damit kann eine maschinell lesbare Lochbandkennzeichnung erfolgen. Rückkehr zur Eingabe 1. KZ durch „:".

## 2.2. Kleines Eingabeprogramm

Das Kleine Eingabeprogramm (KEP) bleibt bei jedem Systemwechsel erhalten und ermöglicht die Eingabe der Teile des jeweiligen Großen Eingabeprogrammes, die zum Einlesen von Triaden mit Kennzeichen, der Leitadressen und der Relativadresse erforderlich sind. Es ermöglicht die Eingabe sowohl im R 300-Code als auch im Fernschreibcode der Zeichen:

0 | 1 | ZR | . | :

Bei der Eingabe des Systems FIPS v muß mit der Eingabe des bitweisen Streifens über die Tastenkombination:

H – G – M↓ – E – E. – M↑ – K – R

begonnen werden (K bedeutet hierbei manuelle Anwahl des benutzten Lesers).

Näheres über das KEP kann den anderen Heften dieser Reihe, z. B. Heft 6 „Gleitkommainterpretationssystem GIPS 1" entnommen werden.

# 3. Systembefehle

Die Liste der durch System FIPS v realisierten Befehle gliedert sich in Standardbefehle, Summenbefehle und Sonderbefehle. Alle Befehle können vor ihrer Abarbeitung modifiziert werden, Summenbefehle allerdings nur mit Einschränkungen.

Jeder Befehl belegt analog den Maschinenbefehlen die Triaden $t_3$–$t_6$ mit der Adresse, in Triade $t_8$ wird stets die Anweisung zur Befehlsmodifikation M verschlüsselt und in $t_7$ dazu die Nummer des Indexregisters angegeben. Für M = 0 oder 4 wird der Befehl vor Abarbeitung nicht verändert.

## 3.1. Befehlsmodifikation

Häufig soll die Abarbeitung eines Befehles, insbesondere dessen Operandenadresse, von bestimmten Parametern abhängen. Eine Änderung der abgespeicherten Befehle ist jedoch sehr unübersichtlich und speicherplatzaufwendig. Eine qualitativ bessere Lösung bietet die Befehlsmodifikation, bei der die Wirkung des Befehles verändert wird, ohne daß der abgespeicherte Befehl beeinflußt wird

Dies erfolgt über die Adressensubstitution, bei der im Befehl nicht die Operandenadresse, sondern die Adresse des Platzes, in der die Operandenadresse als Parameter bereitgestellt wird, angegeben wird, oder über die Adressenänderung, bei der zur programmierten Grundadresse der Inhalt eines der acht sogenannten Indexregister vor Ausführung des eigentlichen Befehles addiert wird. Der Inhalt des Indexregisters kann dabei gleichzeitig für den nächsten Aufruf verändert werden.

Standardbefehle und Sonderbefehle können durch die Befehlsmodifikation jedoch in ihrer Wirkung auch außerhalb der Adressenstellen völlig verändert werden. Insbesondere kann im modifizierten Befehl wiederum eine Befehlsmodifikation angewiesen werden.

Zuerst wird stets eine programmierte Substitution ausgeführt. War außerdem eine Adressenänderung und / oder ein Transport der Adressenstellen zum Indexregister programmiert oder ist die Triade M in der programmierten Zelle entsprechend belegt, werden diese anschließend mit dem neugebildeten Befehlswort ausgeführt.

Bei der Beschreibung der einzelnen Befehle wird außer bei Summenbefehlen stets von der Befehlsstruktur nach Ablauf der Modifikationen ausgegangen, da nur diese für den Ablauf maßgebend ist. In der Strukturdarstellung wird jede Triade durch einen Buchstaben oder eine Ziffer gekennzeichnet, angegebene Ziffern sind bei dem jeweiligen Befehl zwingend vorgeschrieben. B repräsentiert den Inhalt des Pseudobefehlsregisters.

### 3.1.1. Substitution

Codierung: M = 1 oder 3 oder 5 oder 7.

Bei der Substitution wird die im Befehl stehende Operandenadresse durch deren Inhalt ersetzt, wenn in der adressierten Zelle nur die Adressenstellen ungleich Null sind. Sind Triaden auch außerhalb der Adressenstellen in der durch den Befehl bezeichneten Zelle ungleich Null, so werden diese zum Grundbefehl addiert und können so auch weitere Befehlsmodifikationen bewirken.

Ablauf:

a) Löschung der Adressenstellen und des Substitutionsbit im Befehlsregister:

B : = B & 7767 0000 777.

(& bedeutet logische Konjunktion)

b) Addition des Inhaltes der adressierten Zelle:
$B := B + \langle a \rangle$

c) erneute Entschlüsselung des modifizierten Befehles

### 3.1.2. Adressenänderung

Codierung: $M = 2$.

Der Inhalt des Indexregisters I wird zum Grundbefehl addiert. Bleibt dabei Triade $t_{10}$ kleiner 4, wird M gelöscht und zur Befehlsentschlüsselung zurückgekehrt, andernfalls werden $z_0$ und $z_6$ gelöscht und ebenfalls zur Befehlsentschlüsselung zurückgekehrt (Mehrfachänderungen). Sind im angegebenen Indexregister nur Adressenstellen ungleich Null, wird also lediglich diese Adresse zu der programmierten Befehlsadresse addiert. Ein evtl. Überlauf aus dem Adressenbereich ändert nur I (evtl. noch für Sprungbedingung maßgebend) und bleibt im übrigen unberücksichtigt.

Ablauf:

a) Addition des Inhaltes von I in das Befehlsregister:
$B := B + \langle I \rangle$

b) wenn $t_{10}$ kleiner 4, Löschen von M:
B : = B & 7707 7777 777.

c) Löschen von $z_0$ und $z_6$:
B : = B & 3737 7777 777.

d) erneute Entschlüsselung des modifizierten Befehles

### 3.1.3. Registertransport

Codierung: $M = 4$

Der Registertransport stellt zwar keine Befehlsmodifikation dar, steht aber damit in engem Zusammenhang. Es werden die Adressenstellen des (evtl. modifizierten) Befehles in das durch I bezeichnete Indexregister transportiert.

### 3.1.4. Adressenänderung + Registertransport

Codierung: $M = 6$.

Zunächst wird eine Adressenänderung durch Addition des Indexregisterinhaltes ausgeführt. Bleibt dabei die Triade $t_{10}$ kleiner 4, werden die so gebildeten Adressenstellen nach I transportiert, „M" gelöscht und zur Befehlsentschlüsselung zurückgekehrt. Auch bei Überlauf der Adressenstellen erfolgt der Registertransport in das programmierte Indexregister. Wurde $t_{10}$ jedoch gleich oder größer 4, werden außer den Adressenstellen auch die neugebildete Indexregisternummer und $t_{10} = 4$ nach dem programmierten Indexregister transportiert. Da dann vor Rückkehr in das Entschlüsselungsprogramm nur die höchsten bit von $t_8$ und $t_{10}$ gelöscht werden, erfolgt gewöhnlich Mehrfachmodifikation.

Ablauf:

a) Addition des Inhaltes von I in das Befehlsregister:
$B := B + \langle I \rangle$

b) wenn $t_{10}$ gleich oder größer 4, weiter bei f), sonst

c) Registertransport
⟨I⟩ : = B & 0000 7777 000.

d) Löschen der Triade „M":
B : = B & 7707 7777 777.

e) Rückkehr zur Entschlüsselung.

f) Registertransport
⟨I⟩ : = B & 4007 7777 000.

g) Löschen $z_0$ und $z_6$
B : = B & 3737 7777 777.

h) Rückkehr zur Entschlüsselung

### 3.1.5. Einfache Anwendungen der Befehlsmodifikation

a) Gruppenweise Zusammenfassung von Daten auf Grund einer Schlüsselnummer:

Die Schlüsselnummer wird, evtl. mit einem Faktor multipliziert, in die Adressenstellen eines Indexregisters eingegeben. Im Adreßteil des Befehles wird die zur Gruppe 0 gehörige Operandenadresse und in „M" eine 2 (Adressenänderung) programmiert. Die Operation wird mit der um die (evtl. multiplizierte) Schlüsselnummer erhöhten Adresse ausgeführt.

b) Ausführung gleichartiger Operationen (z. B. Löschen) mit einer arithmetischen Folge von Operanden:

Nach erstmaliger Ausführung der Operation in Verbindung mit Registertransport (Transport des Anfangswertes nach I) wird die Operation nochmals, jedoch mit M = 6 (Adressenänderung + Registertransport) und soviel Adresseneinheiten, wie bei jedem Schritt addiert werden sollen, programmiert. Anschließend kann sofort ein Sprung zur erneuten Ausführung dieses Befehles gegeben werden, bis die Operation mit allen Operanden durchgeführt wurde.

Ist die Schrittweite ein Parameter, wird mit M = 7 und Adressierung der Parameterzelle vorher Substitution angewiesen.

c) Rücksprung nach Abarbeitung eines Unterprogrammes nach der in Zelle a gespeicherten Adresse:

Als letzter Befehl des Unterprogrammes wird ein Sprung mit Adresse a und M = 1 ausgeführt. Stellt Zelle a ein Indexregister dar, kann der Rücksprung auch mit M = 2 und a = 0000 programmiert werden.

## 3.2. Standardbefehle

Die Standardbefehle bilden die Grundlage des Systems FIPS v. Es ist möglich, ausschließlich mit Standardbefehlen zu programmieren. Sie sind gekennzeichnet durch $t_{10}$ kleiner 4 und $t_2$ kleiner 4.

In Triade $t_0$ wird die Grundoperation verschlüsselt:

$t_0$ = 0 Leerbefehl
1 Addition
2 Subtraktion
3 Multiplikation
4 Division
5 Eingabe
6 Ausgabe
7 Sprung

Die arithmetischen Operationen stellen Verknüpfungen der Inhalte der Pseudoregister R und AC dar, das Ergebnis steht in AC.

Die Bedeutung der Triaden $t_1$, $t_2$, $t_9$ und $t_{10}$ hängt von der Grundoperation ab. Für $t_0$ = 0 bis 4 ist sie einheitlich.

### 3.2.1. Leerbefehl

Struktur: V S M I $a_1a_2a_3a_4$ W T 0.

Ablauf:

a) Transporte vom Hauptspeicher nach R und AC

T = 0 oder 4 kein Transport
1 oder 5 Transport vom Hauptspeicher nach AC
2 oder 6 Transport vom Hauptspeicher nach R
3 oder 7 Transport vom Hauptspeicher nach AC und R

Die Interpretation des Inhaltes der Zelle a und evtl. a + 1° erfolgt entsprechend der angegebenen Wortlänge W:

W = 0 Füllwort
    1 Miniwort
    2 Normalwort
    3 Doppelwort

b) Verschiebung

Der Inhalt von AC wird um S Dezimalstellen entspr. V verschoben

V = 0 Rechtsverschiebung
    1 Rechtsverschiebung mit Rundung,
      war die größte abgeschnittene Dezimalziffer größer oder gleich 5, wird zur verschobenen Zahl eine 1 addiert.
    2 Linksverschiebung
    3 Linksverschiebung und Betragsbildung

Überlaufanzeige erfolgt, wenn ein Betrag größer $0{,}720575940379279359 \times 10^{18}$ verschoben werden soll oder infolge mindestens zweimaliger Linksverschiebung ein Betrag größer $1{,}152921500580315100 \times 10^{18}$ auftritt.

c) Transport von AC zum Hauptspeicher

T = 0 oder 1 oder 2 oder 3 kein Transport
T = 4 oder 5 oder 6 oder 7 Transport von AC zum Hauptspeicher entspr. W

Wird der durch W vorgegebene Wertebereich überschritten, erfolgt im allgemeinen Überlaufanzeige:

| | | |
|---|---|---|
| 0 Füllwort | $0 \leq AC < 2^{12} = 4{,}096$ | $\times 10^{3}$ |
| 1 Miniwort | $AC < 2^{20} = 1{,}048576$ | $\times 10^{6}$ |
| 2 Normalwort | $AC < 2^{32} = 4{,}294967296$ | $\times 10^{9}$ |
| 3 Doppelwort | $AC < 2^{44} = 1{,}7592186044416$ | $\times 10^{13}$ |

Die Angabe der Pseudoadressen 0000–0007 bewirkt einen Transport von AC um 9 Dualstellen linksverschoben in das durch die Adresse bezeichnete Indexregister. Damit werden in AC stehende Dezimalzahlen in Adresseneinheiten konvertiert. Bei Betrag von AC größer 4096 erfolgt Überlaufanzeige, ein negatives Vorzeichen hat keinen Einfluß. „W" kann dabei beliebig angegeben werden.

Bei Füllwort, Miniwort und Transport mit Pseudoadressen wird kein Überlauf angezeigt, wenn die entsprechenden Beträge ganzzahlige Vielfache von $2^{32}$ sind. Dieser Fall ist jedoch sehr unwahrscheinlich.

Transporte zum Hauptspeicher mit Adressen innerhalb der Befehlsfolge des Systems mit Ausnahme der Pseudoadressen führen zu Überlauf.

Transporte von Normalworten laufen wesentlich rascher als von Füllworten, Miniworten und Doppelworten ab. Ist deshalb bei Transport eines Miniwortes das zugehörige Füllwort noch nicht belegt (z. B. bei Eingaben), wird zweckmäßig der Transport zum Hauptspeicher als Normalwort programmiert.

d) Aufruf des nächsten Befehles

### 3.2.2. Addition

Struktur: V S M I $a_1a_2a_3a_4$ W T 1.

Ablauf:

a) Transporte vom Hauptspeicher nach AC und R gemäß T und W

b) Addition: AC : = AC + R

c) Verschiebung gemäß V und S

d) Transport von AC zum Hauptspeicher gemäß T und W

e) Aufruf des nächsten Befehles

Codierungen und Einzelheiten wie bei Leerbefehl.

### 3.2.3. Subtraktion

Struktur: V S M I $a_1a_2a_3a_4$ W T 2.

Ablauf:

a) Transporte vom Hauptspeicher nach AC und R

b) Subtraktion: AC : = AC — R

c) Verschiebung

d) Transport von AC zum Hauptspeicher

e) Aufruf des nächsten Befehles

Codierungen und Einzelheiten wie der Leerbefehl.

### 3.2.4. Multiplikation

Struktur: V S M I $a_1a_2a_3a_4$ W T 3.

Ablauf:

a) Transporte vom Hauptspeicher nach AC und R

b) Multiplikation: AC : = AC $\times$ R

c) Verschiebung

d) Transport von AC zum Hauptspeicher

e) Aufruf des nächsten Befehles

Codierungen und Einzelheiten wie bei Leerbefehl.

Durch Rechtsverschiebung werden die Kommawerte der Faktoren und des Produktes berücksichtigt:

$S := K_{AC} + K_R - K_P.$

### 3.2.5. Division

Struktur: V S M I $a_1a_2a_3a_4$ W T 4.

Ablauf:

a) Transporte vom Hauptspeicher nach AC und R

b) Verschiebung

Durch die Verschiebung von AC vor Ausführung der Division wird der geforderte Kommawert des Ergebnisses realisiert:

$S := K_Q - K_{AC} + K_R$, für positive S Linksverschiebung.

c) Division AC : = AC / R

Der resultierende Betrag von AC muß kleiner $2^{32}$ sein, sonst erfolgt Überlauf. Divisionsresultate größerer Genauigkeit können durch Rückmultiplikation und Division der Differenz mit größerer Zahl von Linksverschiebungen realisiert werden (Systemunterprogramm).

d) Transport von AC zum Hauptspeicher

e) Aufruf des nächsten Befehles

Codierungen und Einzelheiten wie bei Leerbefehl.

### 3.2.6. Eingabe

#### 3.2.6.1. Eingabe Wort

Struktur: 0 AW M I $a_1a_2a_3a_4$ W T* 5.

Ablauf:

a) Anwahl entsprechend Anwahlinformation

| | | |
|---|---|---|
| AW = | 0 | Anwahl Kanal 0 |
| | 1 | Anwahl Kanal 1 |
| | 2 | Anwahl Kanal 2 |
| | 3 | wirkungslos, bisherige Anwahl bleibt erhalten |
| | 4 | Ausgabe „." auf dem bereits angewählten Kanal |
| | 5 | verboten, |
| | 6 | Ausgabe „Tab" auf dem bereits angewählten Kanal |
| | 7 | Ausgabe „WRZL" auf dem bereits angewählten Kanal |

Im allgemeinen werden bei Eingabe nur die Anwahlinformationen 0–3 sinnvoll sein.

b) Eingabe des Dezimalwortes x nach AC. Beträge größer $1{,}152921488500234239 \times 10^{18}$ führen zu Überlauf. W hat dabei keinen Einfluß.

c) für T* = 3 oder 7 Transport AC nach R

d) Transport AC zum Hauptspeicher wie bei Leerbefehl

e) Aufruf des nächsten Befehles

Wurde statt eines Dezimalwortes eine Marke (darunter wird in diesem Abschnitt „Sm", „Bm", „:", „IS", „IB" oder „'", aber nicht die Wortmarke verstanden) eingegeben, ergeben sich folgende Besonderheiten.

— „WRZL" hat die gleiche Wirkung wie die Zeichenfolge „–."

— „Bm": Die evtl. voranstehenden Ziffern werden als Dezimalzahl nur nach AC eingegeben, R und die programmierte Hauptspeicherzelle bleiben ungeändert. Es wird anschließend nicht der nächste, sondern der übernächste Befehl aufgerufen.

— „:" hat die gleiche Wirkung wie „Bm".

— „IS", „IB" und „'" wirken wie „––", also Löschung des bisher eingegebenen Dezimalwortes und erneuter Eingabebeginn.

#### 3.2.6.2. Eingabe Satz 1

Struktur: 1 AW M I $a_1a_2a_3a_4$ 2 4 5.

Ablauf:

a) Anwahl wie bei Eingabe Wort

b) A : = a

c) Einlesen eines Wortes; wenn Marke, weiter bei f), wenn Betrag größer $4{,}294967289 \times 10^{11}$, Überlauf, sonst

d) Abspeicherung des eingelesenen Wortes als Normalwort in Zelle A

e) A : = A + 1°, weiter bei c)

f) wenn Marke „WRZL": Abspeicherung „0–." in Zelle A, Aufruf des nächsten Befehles,
wenn Marke „Bm": Aufruf des übernächsten Befehles,
wenn Marke „:": Aufruf des übernächsten Befehles
„IS", „IB" oder „'": Rückkehr zu Punkt b)

AC wird durch die Eingabe im Satz verfälscht.

Tritt ein A = I 7 auf, erfolgt Stopp mit Überlaufanzeige.

Die Eingabe im Satz erfolgt schneller als die wortweise Eingabe.

#### 3.2.6.3. Eingabe Satz 3

Struktur: 3 AW M I $a_1a_2a_3a_4$ 2 4 5

Diese Satzeingabe ist mit der Satzeingabe 1 bis auf die Wirkung der Marke „:" identisch.

Ablauf:

a) bis e) s. Pkt. 3.2.6.2.

**f) wenn Marke „WRZL", Abspeicherung „0–." in Zelle A, Aufruf des nächsten Befehles, wenn Marke „Bm", Aufruf des übernächsten Befehles, wenn Marke „IS" oder „IB" oder „'", Rückkehr zu b), wenn Marke „:"**

g) Einlesen eines Wortes

h) Konvertierung in Adresseneinheiten r

i) A : = a + r, weiter bei c).

Mit Satzeingabe 3 können somit Speicherbereiche, die nicht geändert werden sollen, übergangen werden, oder Werte zwecks Korrektur erneut übersprochen werden. Wurde der betreffende Speicherbereich vor der Eingabe gelöscht, brauchen Folgen von Nullen nicht einzeln eingegeben werden.

### 3.2.7. Ausgabe

#### 3.2.7.1. Aufbau Druckregister

Die Druckregister enthalten alle Druckaufbereitungsinformationen. Die Auswahl erfolgt durch die Triade D in jedem Druckbefehl. Der Inhalt des Druckregisters wird wie ein Systembefehl durch 11 Triaden dargestellt.

Aufbau: V S 0 0 $Z_1Z_2$ 0 N 0 0 SZ.

Die Wirkung der Verschiebeinformation V und S stimmt mit dem Leerbefehl überein. Verschiebeanweisungen sind bei Satzausgabe wirkungslos.

Die zweiziffrige Oktalzahl Z gibt die Zahl der Ziffern vor dem Komma an, Vornullen werden durch „ZR" ersetzt. Wird der durch Z festgelegte Wertebereich überschritten, wird die vollständige Zahl, beginnend mit der ersten von Null verschiedenen Ziffer, ausgegeben. Bei der Eingabe kann auch Z = 8 oder 9 angegeben werden. Ab Z = 10 ist jedoch die oktale Verschlüsselung ($Z_1Z_2$) zu beachten.

N (0 . . . 8) gibt die Zahl der Stellen nach dem Komma an. Bei N = 0 wird kein Komma ausgegeben. Vornullen werden eine Stelle vor dem Komma und nach dem Komma als Nullen ausgegeben, bei N = 0 wird mindestens eine Ziffer ausgegeben.

Z + N darf dezimal 17 nicht übersteigen.

Werden Dezimalzahlen nur zur Zwischenspeicherung auf Lochband ausgegeben, wird Z = 0 und N = 0 empfohlen, um redundante Zeichen zu vermeiden.

Am Ende einer negativen Zahl wird grundsätzlich „–" ausgegeben, außerdem kann ein durch SZ codiertes Schlußzeichen ausgegeben werden:

| | | |
|---|---|---|
| SZ = | 0 | Anwahl Kanal 0 |
| | 1 | Anwahl Kanal 1 |
| | 2 | Anwahl Kanal 2 |
| | 3 | wirkungslos |
| | 4 | Ausgabe „." |
| | 5 | Ausgabe „ZR", wenn AC nicht negativ |
| | 6 | Ausgabe „Tab" |
| | 7 | Ausgabe „WRZL" |

### 3.2.7.2. Ausgabe Wort

Struktur: 0 AW M I a a a a W D 6.

Ablauf:

a) Anwahl wie bei Eingabe Wort
b) wenn Adresse a ungleich Null, Transport vom Hauptspeicher nach AC entsprechend „W"
c) Verschiebung wie bei Leerbefehl
d) Ausgabe von AC entsprechend den in D stehenden Druckparametern, AC bleibt erhalten
e) Aufruf des nächsten Befehles

### 3.2.7.3. Ausgabe Satz 1

Struktur: 1 AW M I $a_1a_2a_3a_4$ 2 D 6.

Ablauf:

a) Anwahl wie bei Eingabe Wort
b) A : = a
c) x : = ⟨A⟩
d) wenn x = „0–.", weiter bei g), sonst
e) Ausgabe x entsprechend den in D angegebenen Parametern
f) A : = A + 1°, weiter bei c)
g) Ausgabe „WRZL"
h) Aufruf des nächsten Befehles

AC wird bei Satzausgabe verfälscht.

### 3.2.7.4. Ausgabe Satz 3

Struktur: 3 AW M I $a_1a_2a_3a_4$ 2 D 6.

Ablauf:

a) Anwahl wie bei Eingabe Wort
b) A : = a
c) x : = ⟨A⟩
d) wenn x = „0–.", weiter bei g), wenn x = „0.", weiter bei i), sonst
e) Ausgabe x entsprechend den in D angegebenen Parametern
f) A : = A + 1°, weiter bei c)
g) Ausgabe „WRZL"
h) Aufruf des nächsten Befehles
i) A : = A + 1°
j) x : = ⟨A⟩
k) wenn x = „0.", weiter bei i), sonst
l) Ausgabe „:"
m) x : = A — a mit Konvertierung in eine Dezimalzahl
n) Ausgabe x entsprechend den in D angegebenen Parametern, weiter bei c)

AC wird bei Satzausgabe verfälscht

„Ausgabe Satz 3" dient vor allem der Ausgabe umfangreicher Datenmengen mit zahlreichen Nullen zur Zwischenspeicherung auf Lochband. Ein derartiges Band kann nach Löschung des betreffenden Speicherbereiches mit dem Befehl „Satzeingabe 3" wieder eingelesen werden.

### 3.2.7.5. Ausgabe Text

Struktur: 2 AW M I $\cdot a_1a_2a_3a_4$ 0 0 6.

Ablauf:

a) Anwahl wie bei Eingabe Wort
b) x : = ⟨a⟩
c) Z : = 0
d) $t_9t_8$ von x in Verschlüsselung des Ausgabe-Halt-Befehles konvertieren
e) war $t_9t_8$ = 77, Aufruf des nächsten Befehles, sonst
f) Ausführung Ausgabe-Halt-Befehl
g) Linksverschiebung von x um 6 bit
h) Z : = Z + 1
i) wenn Z kleiner 5, weiter bei d), sonst
j) a : = a + 1°, weiter bei b)

### 3.2.8. Sprungbefehl

Struktur: H 0 M I $a_1a_2a_3a_4$ A B 7.

Ablauf:

a) Prüfen Sprungbedingung B:

B = 0 unbedingt
= 1 AC ≥ 0, aber nicht 0–.
= 2 AC = 0. oder 0–.
= 3 AC ≤ 0 einschließlich 0–.
= 4 Selektor „SP" gesetzt
= 5 ⟨I⟩ = 0
= 6 ⟨I⟩ = 16
= 7 ⟨I⟩ ≤ 16

b) wenn Bedingung erfüllt und Sprungart A = 0 oder 1 oder Bedingung nicht erfüllt und Sprungart A = 2 oder 3, weiter bei d), sonst
c) Aufruf des nächsten Befehles
d) wenn A = 1 oder 3, 17 : = PZ + 1°
e) wenn H = 2, Stopp, wenn Selektor „BS" gesetzt
wenn H = 3, unbedingter Stopp
(werden durch „Start" übergangen)
f) PZ : = a
g) Aufruf des nächsten Befehles PZ

A = 1 oder 3 ermöglicht durch automatische Bereitstellung des um eine Adresseneinheit erhöhten Pseudobefehlszählers in 17 den Aufruf eines Systemunterprogrammes, die Rückkehr in das Hauptprogramm erfolgt durch einen Sprung mit Adresse 0000, M = 2 und I = 7. Wird 17 vom Unterprogramm verändert (z. B. durch Aufruf eines weiteren Unterprogrammes), muß die Rücksprungadresse vorher in eine Arbeitszelle umgespeichert werden. Die Rückkehr in das Hauptprogramm erfolgt dann durch Substitution der Sprungadresse. Die Pseudosprungadresse 0000 führt zum Großen Eingabeprogramm, so daß Programmteile eingelesen oder System-Sofortbefehle abgearbeitet werden können. Ein Anwahlwechsel erfolgt dabei nicht. Die Rückkehr in das gespeicherte Systemprogramm erfolgt z. B. mit 1. KZ „:" (Sprung zu der in 17 gespeicherten Adresse).

Anmerkung:

AC = 0–. entsteht niemals durch Rechenoperationen (Ausnahme: Addition 0–. + 0–.) oder Rechtsverschiebungen. Der Wert „0–." kennzeichnet intern die Satzmarke.

## 3.3. Summenbefehle

Summenbefehle erlauben nicht nur die Einsparung von Befehlen durch Beibehaltung des Inhaltes von R und AC, sondern laufen vor allem wesentlich rascher als die entsprechenden Standardbefehle ab. Summenbefehle werden nur mit Normalworten ausgeführt. Als Befehlsmodifikation ist nur die Adressenänderung zugelassen, und diese wirkt ausschließlich auf die Adressenstellen.

### 3.3.1. Summenbefehl Addition

Codierung: 4 0 M I $a_1a_2a_3a_4$ 2 4 1.

Ablauf:

a) wenn M $\geq$ 2, $a^* := a + \langle I \rangle$; sonst $a^* := a$

b) $\langle a^* \rangle := \langle a^* \rangle + AC$

c) Aufruf des nächsten Befehles

### 3.3.2. Summenbefehl Subtraktion

Codierung: 4 0 M I $a_1a_2a_3a_4$ 2 4 2.

Ablauf:

a) wenn M $\geq$ 2, $a^* := a + \langle I \rangle$; sonst $a^* := a$

b) $\langle a^* \rangle := \langle a^* \rangle - AC$

c) Aufruf des nächsten Befehles

## 3.4. Sonderbefehle

Sonderbefehle werden nach der üblichen Befehlsmodifikation auf Maschinenbefehle zurückgeführt. Sie können mit dem Maschinenakkumulatorinhalt „0" oder als Verknüpfungen der Inhalte der Indexregister mit den adressierten Operanden angewiesen werden. Eine Interpretation der Operanden als Dezimalzahlen (Normalworte) erfolgt nur für positive Werte, da Maschinenbefehle eine Verschlüsselung negativer Werte in Komplementdarstellung voraussetzen.

Sonderbefehle sind durch $t_2$ größer oder gleich 4 gekennzeichnet. Sie dürfen die Spuren 7540–7577 und 7700–7777 als Arbeitsspuren benutzen. Der Inhalt dieser Spuren wird durch den Aufruf weiterer Sonderbefehle nicht verändert.

### 3.4.1. Sonderbefehle ohne Beeinflussung der Indexregister

Struktur: 0 0 M I $a_1a_2a_3a_4$ $b_5b_6b_7$.

Ablauf:

a) b : = B — 0000 0000 300.

b) ac : = 0

c) Ausführung b

d) Aufruf des nächsten Befehles

(b = Maschinenbefehl, ac = Inhalt Maschinenakkumulator)

### 3.4.2. Sonderbefehle mit Beeinflussung der Indexregister

Struktur: 2 $I^+$ M I $a_1a_2a_3a_4$ $b_5b_6b_7$.

Ablauf:

a) b : = B — 0000 0000 300.

b) ac : = $\langle I^+ \rangle$

c) Ausführung b

d) $\langle I^+ \rangle$ : = ac

e) Aufruf des nächsten Befehles

### 3.4.3. Liste ausgewählter Sonderbefehle

Die Anwendung nachstehender Sonderbefehle ist ohne Kenntnis der Maschinensprache möglich.

#### 3.4.3.1. Stoppbefehle

0 0 M I $a_1a_2a_3a_4$ 4 2 0. unbedingter Stopp

0 0 M I $a_1a_2a_3a_4$ 4 6 0. Stopp, wenn Selektor „BS" gesetzt

Da bei diesen Stopp-Befehlen im Gegensatz zu den Stoppanweisungen im Standard-Sprungbefehl die Adresse in der Operationsregisteranzeige (Lampenfeld) angezeigt wird, ist eine Kennzeichnung der einzelnen Stopp-Stellen möglich. Die Adresse a hat keinen weiteren Einfluß, die Programmabarbeitung wird mit „Start" fortgesetzt.

#### 3.4.3.2. Löschbefehle

0 0 M I $a_1a_2a_3a_4$ 5 4 0. Löschen Zelle

0 0 M I $a_1a_2a_3a_4$ 7 4 0. Löschen Spur

Eine Spur beinhaltet die Adressen xx00–xx37 oder xx40–xx77. In „Löschen Spur" kann eine beliebige Zelle der Spur angegeben werden, es wird stets die gesamte Spur gelöscht.

#### 3.4.3.3. Ausgabe Zeichen

0 0 M I $a_1a_2a_3a_4$ 4 3 6.

a stellt bei diesem Befehl keine Adresse, sondern die Schlüsselnummer des auszugebenden Zeichens dar. Der Befehl kann auch zur Anwahl eines Kanales benutzt werden:

a = 4700 Anwahl Kanal 0
4600 Anwahl Kanal 1
4500 Anwahl Kanal 2

Bei Ausgabe eines Zeichens erfolgt kein vorheriger Anwahlwechsel. Schlüsselliste siehe Tab. 1.

#### 3.4.3.4. Transport Indexregistereinheit nach a

2 $I^+$M I $a_1a_2a_3a_4$ 5 4 0.

Der Inhalt des Indexregisters $I^+$ wird nach Zelle a transportiert, der Indexregisterinhalt bleibt erhalten.

#### 3.4.3.4. Transport Indexregisterinhalt nach a

2 $I^+$M I $a_1a_2a_3a_4$ 5 1 1.

Der Inhalt von a wird nach $I^+$ transportiert.

#### 3.4.3.6. Addition Spur

2 $I^+$M I $a_1a_2a_3a_4$ 7 1 1. $\langle I^+\rangle$ : = Summe Spur

2 $I^+$M I $a_1a_2a_3a_4$ 7 1 0. $\langle I^+\rangle$ : = Summe Spur + $\langle I^+\rangle$

Bei Auftreten negativer Zahlen oder bei Überlauf (dessen Anzeige unterdrückt wird) erfolgt nur eine formale Addition, z. B. zur Bildung eines Kontrollwertes.

## 3.5. Übergang von der Abarbeitung von Systembefehlen zu Maschinenbefehlen und umgekehrt

Aus Zeitgründen ist oft die Einfügung von Maschinenbefehlsfolgen in Systemprogramme zweckmäßig. Besonders trifft das auf die Verwendung vorliegender Standardunterprogramme (z. B. für elementare mathematische Funktionen) zu. Werden die Besonderheiten der Parameterbereitstellung des betreffenden Unterprogrammes beachtet, kann dieses mit dem Sonderbefehl

0 0 M I $a_1a_2a_3a_4$ 4 2 7.

aufgerufen werden. Nach Abarbeitung führt das Unterprogramm zwangsläufig wieder in das Systemprogramm zurück. Soll der Maschinenakkumulatorinhalt ac nach Rücksprung in einem Indexregister $I^+$ gespeichert werden, muß

2 $I^+$M I $a_1a_2a_3a_4$ 4 2 7.

programmiert werden.

Wird das Maschinen-Unterprogramm nur von Systembefehlen aufgerufen, kann die Abspeicherung der Rücksprungadresse im UP entfallen, wenn als letzter Maschinenbefehl ein Sprung nach Zelle 7604 programmiert wird.

Ein Übergang von Systembefehlen zu Maschinenbefehlen ist auch mit dem Sonderbefehl

0 0 M I $a_1a_2a_3a_4$ 4 0 7.

möglich.

Steht dann als letzter Maschinenbefehl in Zelle c: 0331 025., wird ab Zelle c + 1° die Abarbeitung der Systembefehle fortgesetzt.

Die Maschinenprogramme dürfen die Arbeitsspuren 7540–7577, 7640–7677 und 7700–7737 des Systemes benutzen.

Als Maschinenunterprogramm stehen in FIPS v zur Verfügung:

| | Ansprungzelle | Rücksprungzelle |
|---|---|---|
| Eingabe Oktalwort nach ac | 0422 | 7745 |
| Eingabe Adresse nach ac | 0263 | 7655 |
| Eingabe Dezimalwort nach ac (Grenzwert wie bei 1. KZ = 3) | 0713 | 7745 |

Weitere UP würden eine unrationelle Parameterversorgung bedingen und sind deshalb nicht vorgesehen. Durch ein universelles Unterprogramm

```
0000 00 7602 270.  PZ : = 0
  01      >1°< 210.  RS + 1°
  02 00 7750 240.
  03      >1°< 220.  RS
  04 00  335 000.
```

wird statt dessen nur das auf den UP-Aufruf folgende Wort als Systembefehl interpretiert (Eingabe im Satz ist dabei nicht zugelassen, bei Eingabe Wort darf „Sm“ nicht auftreten). Bei wiederholtem Aufruf ohne dazwischenliegenden Übergang zur interpretierenden Abarbeitung kann das universelle UP bei Platz 0001 angesprungen werden. Das universelle UP kann auf beliebige freie Plätze gespeichert werden.

Damit können Systembefehle in ein Maschinenprogramm eingebaut werden.

## 3.6. Programmierbeispiele

### 3.6.1. Bestandsrechnung

Es sind von einem Lochstreifen jeweils Artikelgruppennummer, Menge (2 Kommastellen) und Einzelpreis (3 Kommastellen) einzulesen. Mengen und ausgerechnete Werte (2 Kommastellen) sind getrennt nach Artikelgruppen und insgesamt zu saldieren. Das Lochband wird mit einer Blockmarke abgeschlossen, danach sind die Summen je Artikelgruppe mit Satzausgabe 3 auf Lochband auszustanzen. Anschließend ist eine formale Kontrollsumme durch Spuraddition zu bilden und ebenfalls auszustanzen.

Vor Beginn des Arbeitsganges wird das entsprechende Lochband des Vortages eingelesen. Die Kontrollsumme ist zu prüfen, bei Fehler Stopp.

Die Gesamtsummen Mengen und Werte sollen als Doppelworte ständig gespeichert bleiben.

Die Gruppennummern liegen im Bereich 0–509. Wird eine größere Nummer erkannt, erfolgt deren Ausdruck und Stopp. Nach „Start" werden Menge und Wert einer Sammelnummer 510 zugerechnet.

Das Programm ist relativ programmiert, der Beginn des Summenbereiches (Anfangsadresse $A_0$) wird mit der 2. Leitadresse bezeichnet.

Die Gesamtsummen werden relativ zur 3. Leitadresse auf den Plätzen 0 und 2 gespeichert.

a) Löschen Speicherbereich vor Satzeingabe 3:

```
0000  02     41 0000 740.  I 1 : = A°, Löschen 1. Spur
0001  02     46 1740 000.  I 6 : = A + 1740° (dez. 992)
0002  00     61 0040 740.  I 1 : = I 1 + 40°, Löschen Spur i
0003  01      1 0002 267.  Sprung, wenn nicht I 1 = I 6
```

b) Eingabe Satz

```
0004  02 3100 0000 245.  Satzeingabe 3 über Leser 1
```

c) Bildung Kontrollsumme

```
0005  02 2241 0000 711.  I 1 : = A°, I 2 : = Summe Spur
0006  00 2261 0040 710.  I 1 : = A + 40°,
                         I 2 : = I 2 + Summe Spur
0007  01      1 0006 267.  Sprung, wenn nicht I 1 = I 6
```

d) Einlesen Kontrollsumme und Vergleich

```
0010  00    100 7766 255.  I 6 : = Kontrollsumme von Leser 1
0011  01 3002 0000 267.  Sprung mit Stopp, wenn nicht I 2 = I 6
```

e) Bereitstellung max. Gruppennummer

```
0012  00     46 0775 000.  I 6 : = 775° (dez. 509)
```

f) Einlesen Gruppennummer,

```
0013  00    200 0001 255.  Eingabe Nr. über Leser 2 nach I 1
0014  01          24 007.  Sprung, der durch „Bm" übergangen
                           wird
```

g) Ausdruck und Korrektur bei Nr. größer 509

```
0024  01      1 0027 077.  Sprung, wenn Nr. ≤ 509
0025  00    700 0000 016.  Ausgabe Nr. (nach WRZL)
0026  00     41 0776 460.  bed. Stopp, Korrektur Nr. : = 510
```

h) Eingabe Menge und Wert mit Summenbildung

| | | | | | |
|---|---|---|---|---|---|
| 0027 | 00 | 200 | 0000 | 035. | Eingabe Menge M nach AC u. R |
| 0030 | 02 | 4021 | 0000 | 241. | M (Nr.) : = M (Nr.) + M |
| 0031 | 03 | | | 351. | $M_{ges} := M_{ges} + M$ |
| 0032 | 00 | 200 | 0000 | 015. | Eingabe Preis nach AC |
| 0033 | 00 | 1300 | 0000 | 003. | AC : = P × M, 3 Stellenabstreichungen, kaufm. Rundung |
| 0034 | 02 | 4021 | 1000 | 241. | W (Nr.) : = W (Nr.) + AC |
| 0035 | 03 | | 2 | 361. | $W_{ges} := W_{ges} + W$ |
| 0036 | 01 | | 13 | 007. | Sprung zum Einlesen nächste Nr. |

i) Ablauf nach Eingabe „Bm"

| | | | | | |
|---|---|---|---|---|---|
| 0013 | 00 | 200 | 0001 | 255. | Eingabe „Bm", weiter bei übernächstem Befehl |
| 0015 | 02 | 3100 | 0000 | 206. | Ausgabe Satz 3 auf Lochband |
| 0016 | 02 | 2241 | 0000 | 711. | I 1 : = A°, Summieren 1. Spur in I 2 |
| 0017 | 02 | 0046 | 1740 | 000. | I 6 : = A + 1740° |
| 0020 | 00 | 2261 | 0040 | 710. | I 1 : = I 1 + 40°, summieren i.-te Spur in I 2 |
| 0021 | 01 | | 1 0020 | 267. | Sprung, wenn nicht I 1 = I 6 |
| 0022 | 00 | 100 | 7762 | 206. | Ausgabe Kontrollsumme auf LB |
| 0023 | 01 | 3000 | 0000 | 007. | unbedingter Stopp, nach „Start" erneuter Ablauf des Programmes (z. B. nächste Abteilung) |

Die verwendeten Druckregister haben folgende Belegung:

| | | | | |
|---|---|---|---|---|
| 7630 | 00 | | 4. | Ausgabe ohne „ZR" und Komma, SZ = „." |
| 31 | 00 | 300 | 003. | Ausgabe 3 Stellen ohne Komma und SZ |

Im Anhang ist die Befehlsfolge noch einmal zusammenhängend dargestellt.

### 3.6.2. Toleranzprüfung

Im Automaten werden fortlaufend ab 2. Leitadresse Sollinformationen gespeichert. Auf weiteren Lochbändern stehen zugehörige Istwerte (gleiche Kommawerte). Positionen, die nicht belegt sind, werden durch „:" und Angabe der Nummer der nächstfolgenden Position gekennzeichnet.

Die Bänder werden mit „Sm" abgeschlossen.

Ein Ausdruck von Soll- und Istwert, Prozentsatz und Positionsnummer erfolgt bei Abweichungen größer n % (Genauigkeit 1 Kommastelle, n = Parameter, gespeichert in I 6).

Arbeitsspeicher ab 3. Leitadresse

| | | | |
|---|---|---|---|
| 0000 | 02 | 1100 0000 245. | Eingabe Sollwerte (Satz 1) |
| 01 | 00 | 7761 540. | I 1 : = 0 |
| 02 | 03 | 200 0000 255. | Eingabe Istwert oder „:" |
| 03 | 01 | 6 007. | Sprung nach Eingabe Istwert |
| 04 | 00 | 200 0001 255. | I 1 : = neue Nr. (Adresseneinheit) – Fortsetzung nach Eingabe „:" – |
| 05 | 01 | 2 007. | |
| 06 | 01 | 24 027. | P (AC = 0) |
| 07 | 02 | 21 0000 222. | D : = Istwert — Sollwert |
| 10 | 00 | 3300 7762 244. | d : = \|D / Sollwert\|, 3 KoSt |
| 11 | 01 | 2 0022 077. | P (d ≤ n) |
| 12 | 02 | 721 0000 216. | Druck Sollwert nach WRZL |
| 13 | 03 | 300 0000 216. | Druck Istwert |
| 14 | 00 | 2300 0000 004. | Istwert / Sollwert, 3 KoSt |
| 15 | 00 | 300 0000 026. | Druck Prozent |
| 16 | 00 | 1740 436. | Druck „%" |
| 17 | 01 | 26 020. | R : = $2^9$ (als Füllwort) |
| 20 | 00 | 7761 214. | AC : = I 1 / $2^9$ |
| 21 | 00 | 300 0000 026. | Druck Nr. |
| 22 | 00 | 61 0001 000. | I 1 : = I 1 + 1° |
| 23 | 01 | 2 007. | |
| 24 | 01 | 7 017. | P (AC = 0 +.) |
| 25 | 01 | 3000 0000 007. | Stopp bei Sm |
| 26 | 3 | 123456—. | Eingabe Miniwort für anderen Programmteil als Normalwort |
| | 51 | 100 0026 055. | Eingabebefehl für Füllwort nach h + 26°. |
| | | 512. | = $2^9$ |

Druckregister:

| | | | |
|---|---|---|---|
| 7631 | 00 | 1200 006. | 10 Stellen ohne Komma, SZ = „Tab" |
| 32 | 00 | 301 003. | 3 Stellen vor, 1 Stelle nach dem Komma, SZ wirkungslos |

### 3.6.3. Inventurauswertung

Je Position soll Karteibestand, aufgenommener Bestand, Artikel-Nr. und bei Abweichung die Schlüsselnummer der Ursache im Satz eingelesen werden. Es sind, getrennt nach Schlüsselnummern der Ursache und Artikelgruppen, die Zahl der Positionen mit Abweichungen und die Differenzen zu summieren. Die Artikelgruppe ergibt sich aus der Artikelnummer durch Abstreichen der zwei letzten Stellen. Infolge des vorgegebenen Wertebereiches sind die Differenzsummen als Doppelworte und die Positionszahlen als Miniworte speichern. Der gespeicherte Bereich ist nach Eingabe aller Daten zu stanzen. Vor Beginn der Arbeit ist der entsprechende Streifen des vorhergehenden Durchlaufes einzugeben. Ab 2. LA sind die Summen zu speichern, die Arbeitszellen beginnen ab 3. LA. Die 4. LA gibt den Beginn des Speicherbereiches Artikelgruppe (AG) an. Das Ende des Summenspeicherbereiches ist durch die 6. LA (= I 6) anzugeben.
Anmerkung: Dient der Lochstreifen nur der Zwischenspeicherung, kann ein beliebiger Speicherinhalt formal als Folge von Normalworten interpretiert und entsprechend schneller ausgestanzt (auch durch Satzausgabe) und wieder eingelesen (nur mit „Eingabe Wort" wegen Überlauf) werden, doch ist der Inhalt des Streifens dann nur durch den C 8205 lesbar. Hier wurde deshalb davon kein Gebrauch gemacht.

| | | | | |
|---|---|---|---|---|
| 0000 | 02 | 141 0001 | 255. | I 1 : = A + 1°, Einlesen 1. Wort |
| 01 | 00 | 321 7777 | 355. | Eingabe Doppelwort |
| 02 | 00 | 361 0002 | 255. | I 1 : = I 1 + 2°, Eing. Miniwort als Normalwort oder Bm |
| 03 | 01 | 1 | 007. | |
| 04 | 03 | 47 0005 | 000. | I 7 : = Grenzwert für Eing. Satz |
| 05 | 03 | 1200 0000 | 245. | Eingabe Satz oder Bm |
| 06 | 01 | 16 | 007. | |
| 07 | 02 | 141 0001 | 106. | I 1 : = A + 1°, Ausgabe 1. Miniwort |
| 10 | 02 | 100 0000 | 306. | Ausgabe 1. Doppelwort |
| 11 | 00 | 361 0002 | 106. | I 1 : = I 1 + 2°, Ausgabe Miniwort |
| 12 | 00 | 321 7777 | 306. | Ausgabe Doppelwort |
| 13 | 01 | 1 0011 | 077. | P (I 1 ≦ I 6) |
| 14 | 00 | 3520 | 436. | Ausgabe Bm |
| 15 | 01 | 3000 0000 | 007. | Stopp |
| 16 | 03 | 0 | 220. | R : = Karteibestand |
| 17 | 03 | 1 | 252. | D : = aufgen. Best. – Karteibest. |
| 20 | 01 | 5 | 027. | P (D = 0) |
| 21 | 03 | 3 | 231. | AC : = 2 Nr. |
| 22 | 00 | 1 | 240. | I 1 : = 2 Nr. (Adr. Einh.) |
| 23 | 03 | 200 0002 | 210. | AG : = Artikelnummer/100 |
| 24 | 00 | 7610 | 221. | AC : = 2 AG (in 7610 steht der Betrag von AC kleiner $2^{32}$) |
| 25 | 00 | 2 | 240. | I 2 : = 2 AG (Adr. Einh.) |
| 26 | 03 | 1 | 220. | R : = D |
| 27 | 02 | 21 0000 | 351. | D (Nr.) : = D (Nr.) + D |
| 30 | 04 | 22 0000 | 351. | D (AG) : = D (AG) + D |
| 31 | 01 | 35 | 220. | R : = 1 |
| 32 | 02 | 21 0001 | 151. | Z (Nr.) : = Z (Nr.) + 1 |
| 33 | 04 | 22 0001 | 151. | Z (AG) : = Z (AG) + 1 |
| 34 | 01 | 5 | 007. | |
| 35 | 3 | | 1. | |

Druckregister:

| | | | |
|---|---|---|---|
| 7630 | 00 | | 4. |

# 4. Hilfsprogramme

Die Hilfsprogramme werden zur Fehlersuche beim Testen von Programmen, zum Vergleichen, Programmausschreiben und -stanzen benötigt. Bei der Abarbeitung erprobter Programme können sie z. B. als Arbeitszellen überschrieben werden.

## 4.1. Ausdruck R dezimal

Mit den KZ „D 0" wird R dezimal entsprechend den im Druckregister 0 angegebenen Parametern ausgedruckt. AC muß sich als Doppelwort zwischenspeichern lassen (höchstens 13 Dezimalstellen). Anschließend ist mit der Eingabe 1. KZ über Tastatur fortzufahren.

Hilfsprogramme werden ab 7400 benötigt.

## 4.2. Ausdruck PZ und AC

Die KZ „D 2" bewirken einen Ausdruck der Adresse, die im Pseudobefehlszähler (Zelle 7600) gespeichert ist, und des AC ohne Zwischenraum und Komma. Wird anschließend eine Wartezelle w. eingetastet, wird das Pseudoprogramm, bei PZ beginnend, bis zur Wartezelle w abgearbeitet.

Rückkehr zur Eingabe 1. KZ durch H – G – R.

Hilfsprogramme werden ab 7500 benötigt.

## 4.3. Bereichsausschreiben

Bei nicht gesetztem Selektor „SP" wird, beginnend mit der nach den Kennzeichen eingegebenen Adresse, fortlaufend die Adresse der Zelle und deren Inhalt ausgedruckt. Bei gesetztem Selektor wartet der Automat nach vollständigem Ausdruck auf die Eingabe einer neuen Anfangsadresse. KZ „D 3" bewirkt den oktalen Ausdruck des Zelleninhaltes, bei „D 4" muß nach der Adresse (ohne Wortmarke) noch W und D angegeben werden, erst dann folgt die Wortmarke. Der Zelleninhalt wird dann entsprechend W als Dezimalzahl interpretiert und mit den im Druckregister D stehenden Parametern ausgedruckt. Bei W = 3 (Doppelwort) wird natürlich nur jeder zweite Wert sinnvoll sein.

Rückkehr zur Eingabe 1. KZ mit H – G – R (vgl. 6.11.).

Für „D 3" muß ab 7440, für „D 4" ab 7400 der Teil „Hilfsprogramme" abgespeichert sein.

## 4.4. Ausdruck Zelleninhalt 7777 bzw. bz und ac

Mit KZ „D 1" wird zunächst eine belanglose Adresse und anschließend der Inhalt der Zelle 7777 oktal ausgedruckt. Dies ist nur bei Abarbeitung von Maschinenbefehlen interessant (siehe „Anleitung zur Programmierung in der Maschinensprache). Rückkehr zur Eingabe 1. KZ. Soll zu einem beliebigen Zeitpunkt Maschinenbefehlszähler bz und Maschinenakkumulatorinhalt ac ausgedruckt werden, so ist mit H – G – Ü – E zunächst der ac nach Zelle 7777 zu transportieren und anschließend mit G – M↓ – E – M↑ – R das Ausdruckprogramm für bz und ac anzuspringen. Nach einem bedingten Stopp wird die Programmabarbeitung fortgesetzt.

Die Hilfsprogramme müssen ab Zelle 7500 gespeichert sein.

## 4.5. Vergleichsprogramm

Das Vergleichsprogramm erfordert eine Eingabe mit Kennzeichen wie das Große Eingabeprogramm. Worte mit den 1. KZ 0, 3, 4, 8 oder

Punkt werden jedoch nicht abgespeichert, sondern mit dem abgespeicherten Inhalt verglichen. Bei Abweichungen erfolgt Ausdruck der Adresse, des eingelesenen und des abgespeicherten Inhaltes (letzterer nur bei Inhalt ungleich Null). Da anschließend vor Fortsetzung der Eingabe kein Anwahlwechsel erfolgt, ist Leser 2 zu benutzen. System-Sofortbefehle werden ausgeführt.

KZ „D 5" bewirkt oktalen und „D 6" dezimalen Ausdruck (Normalworte entsprechend den in „D 0" stehenden Parametern).

Benötigte Befehle: ab 7440 für oktalen Ausdruck
ab 7400 für dezimalen Ausdruck

### 4.6. Bereichsausstanzen

Eingabebild: D 8 a. e.

Es werden „21 a. 40" und triadenweise alle Zelleninhalte von a bis e–1', anschließend „:" und „/" ausgestanzt. Somit können beliebige Speicherbereiche so ausgegeben werden, wie sie vom Großen Eingabeprogramm wieder eingelesen werden (Oktalzahlen im Satz). Rückkehr zur Eingabe 1. KZ. Das Programm ist ab Zelle 7340 gespeichert.

# 5. Herstellung eines Lochbandes

Dieser Abschnitt vermittelt keine neuen Kenntnisse. Es werden nur Hinweise zum Ablochen von Lochbändern zusammengestellt.

## 5.1. Allgemeines

— Jedes Wort wird durch eine Wortmarke abgeschlossen, wenn es nicht selbst eine Marke darstellt.

— Innerhalb des Wortes können Ziffern, die wirkungslosen Zeichen (, ZR GB KB SPE IZ), die einer Null gleichwertigen Zeichen (WR NS WS + Leserstopp) und unmittelbar vor der Wortmarke ein negatives Vorzeichen stehen.

— Die Zeichenfolge „−−−" löscht den bereits eingegebenen Wortteil (darf nicht bei Leitadressen verwendet werden).

— Transportlochung ohne Informationslochung führt außerhalb des Vorspannes zu „Fehler Eingabe".

## 5.2. Programmstreifen

— Die Adresse der Zelle, in die abgespeichert werden soll, wird nicht abgelocht.

— Die Eingabe von Worten oder Sätzen, bestehend aus einer Folge von Worten und einer Endmarke, wird durch die vorangestellten Kennzeichen, die selbst nicht mit abgespeichert werden, gesteuert.

— Die Abspeicherung erfolgt jeweils nach „Wortmarke" in aufeinanderfolgenden Zellen, solange keine 1. Leitadresse oder Relativadresse abgelocht wird.

— Das Zeichen „:" bzw. „Bm" außerhalb der Kennzeichen führt stets zur Eingabe 1. KZ ohne Veränderung des Abspeicherbefehles und wird insbesondere als Endmarke bei Satzeingabe verwendet.

— „Sm" (WRZL) bewirkt außerhalb der Kennzeichen eine Abspeicherung von „0−." und führt zur Eingabe 1. KZ.

— Durch Angabe einer Relativadresse können bereits abgelochte Programmteile übersprochen werden, bei Vergleich erfolgt dann jedoch zunächst Ausdruck.

— Sätze von Dezimalzahlen können vor Ablochung der Endmarke durch „IS", „IB" oder „′" gelöscht werden, bei Vergleich erfolgt dann jedoch evtl. Ausdruck.

— „ZR" und „IZ" sind auch als Kennzeichen wirkungslos.

— Nicht belegte 1. KZ führen zur erneuten Eingabe 1. KZ. So können vor Ablochung jedes 1. Kennzeichens die Wagenoperationen Tabulator, WRZL, WR, ZL unbedenklich ausgeführt werden.

— System-Sofort-Befehle werden auch bei Programmvergleich ausgeführt.

— Eingaben durch System-Sofort-Befehle zerstören den Abspeicherbefehl, die folgende Dezimalzahl unterliegt den Regeln für Daten.

— 1. KZ „:" führt zum Verlassen der Programmeingabe (Ansprung Systemprogramm bei I 7 = 7. LA).

— 1. KZ = 1 führt zum Verlassen der Programmeingabe, ebenso 1. KZ = 5 (System-Sofort-Befehl), wenn ein Sprungbefehl folgt.

## 5.3. Datenstreifen

— Datenstreifen bestehen aus Dezimalworten oder Sätzen von Dezimalworten und enthalten keine Kennzeichen.

— Sätze von Dezimalzahlen können vor Ablochung der Endmarke nur dann durch Ablochung von „IS", „IB" oder „′" gelöscht werden, wenn der Satz nur mit den Befehlen „Eingabe im Satz 1" oder „Eingabe im Satz 3" aufgerufen wird.

— Wird ein Satz nur mit „Eingabe im Satz 3" aufgerufen, können fehlerhafte Werte vor Lochung der Satzmarke auch mit „:a." (a = Nummer des fehlerhaften Platzes dezimal, die Zählung beginnt mit 0) und richtiger Ablochung korrigiert werden. Anschließend muß entweder das darauffolgende Wort oder erneut „:a." gelocht werden, nicht aber sofort die Satzmarke.

— „Sm" = WRZL beendet einen Dezimalsatz, und es wird gleichzeitig als letztes Wort „0–." abgespeichert.

— „Bm" beendet einen Satz mit anderer Wirkung im Programm als „Sm".

— „·" darf nicht einen Satz abschließen, wenn dieser von einem Befehl „Eingabe im Satz 3" aufgerufen wird. In den anderen Fällen ist er der Blockmarke gleichwertig.

— Ziffern ungleich Null unmittelbar vor der Satzmarke führen zu fehlerhaftem Programmlauf, wenn diese Marke durch „Eingabe Wort" eingelesen wird.

# 6. Allgemeine Regieanweisung

Diese Regieanweisung soll durch Zusammenstellung bereits erläuterter Fakten Hinweise zur Arbeit mit dem System FIPS v am Automaten geben. Sie ist durch spezielle Regieanweisungen zu den einzelnen Programmen zu ergänzen.

## 6.1. Systemeingabe

a) H – G (Schalter „Halt" einrasten, generallöschen)

b) Bahnsperren außer Spur 0 öffnen

c) Einlegen bitweiser Streifen (0–277) in Leser i (i = 1 oder 2)

d) M↓ – E – E – M↑ – Ki – R (Ki = Anwahltaste i betätigen) bitweiser Streifen wird eingelesen, anschl. Anzeige „FE"

e) H – G – R – H

f) Einlegen Systemstreifen (0–2277, 7340–7627) in Leser i

g) (evtl.) Setzen Selektor „BS"

h) Ki – R Streifen wird eingelesen
wenn „BS" gesetzt, Stopp nach Eingabe 0–2277

i) sollen Hilfsprogramme eingelesen werden, lösen „bS", sonst H – G – R

k) Bahnsperren schließen

nach Einlesen des gesamten Streifens:

> Automat steht im Grundzustand „Eingabe 1. KZ über Tastatur". Dieser Zustand kann bei aufgesprochenem System stets durch „H – G – R" erreicht werden und wird bei manuellen Eingriffen im allgemeinen vorausgesetzt.

## 6.2. Systemkontrolle

B 1 Nach Eingabe der Kennzeichen B 1 rechnet der Automat einige Sekunden und geht bei einwandfrei abgespeichertem System wieder zur Eingabe 1. KZ, bei Fehler „Stopp". Zur Lokalisierung der Fehler kann der Systemstreifen mit Vergleichsprogramm eingelesen werden, unterhalb 2277 dürfen dabei nur die Arbeitszellen 2140 und 2147 ausgedruckt werden.

Erfolgen keine weiteren Ausdrucke trotz Fehleranzeige, ist der Systemstreifen beschädigt oder der Automat nicht funktionssicher.

Nach Ausschreiben fehlerhafter Zellen manuelle Korrektur über die Tastatur nach Öffnen der entsprechenden Bahnsperren (siehe Bedienungsanleitung) oder erneute Systemeingabe.

## 6.3. Ausschalten von „Textausgabe" bzw. „Division"

Werden die Systembefehle „Textausgabe" bzw. „Division" nicht benötigt, können die entsprechenden Änderungsstreifen eingelegt werden. Die Eingabe erfolgt nach 6.1. b), e), h) und k). Treten danach diese Befehle im Programm auf, wird Überlauf angezeigt. Die Systemkontrolle wird den geänderten Bedingungen angepaßt. Ohne Textausgabe kann ab 2240, ohne Division ab Zelle 2100 und ohne beide Befehle ab Zelle 2040 frei bis zum Hilfsprogrammteil bzw. bis 7537 verfügt werden.

Werden die obengenannten Befehle später wieder benötigt, braucht nur der Spezialstreifen „2000–2277 + Kontrolle" wie oben eingelesen werden. Das gleiche gilt bei versehentlicher Zerstörung dieser Plätze. In letzterem Fall brauchen jedoch die Bahnsperren nicht geöffnet werden.

## 6.4. Programmeingabe

Bei der Programmeingabe unterscheidet man zwischen absoluten Streifen, die alle Leitadressen enthalten und damit bezüglich des Speicherbereiches genau festgelegt sind, und relativen Streifen, bei denen durch manuelle Eingabe der Leitadressen der Speicherbereich für Hauptprogramm, Konstanten, Datenfelder und Unterprogramme erst zugeordnet wird. Vom Rechner gestanzte Programme sind stets absolut, während sie bei der Programmierung im allgemeinen relativ geschrieben werden. Sie können aber bereits vor dem Ablochen durch Zuweisung der Leitadressen zu absoluten werden.

### 6.4.1. Eingabe absolutes Programm

a) Einlegen Lochstreifen in Leser i

b) H – G – R – H – Ki – R

Programm wird eingelesen, nach Eingabe 1. KZ „/" bedingter Stopp, bei 1. KZ „e" Übergang zur manuellen Eingabe 1. KZ. „BS" kann durch Lösen des Selektors oder durch „Start" übergangen werden.

### 6.4.2. Eingabe relatives Programm

a) H – G – R

21 HLA. Eingabe 1. Leitadresse (Hauptleitadresse)
22 NLA. Eingabe 2. bis 7. Leitadresse (Nebenleitadressen)
.
.
.
27 NLA.

Es ist der maximal verfügbare Speicherraum zu beachten!

b) weiter wie bei absolutem Programm

Treten auf dem Programmstreifen Sofort-System-Befehle zur Eingabe von Füllwörtern, Miniwörtern und Doppelwörtern auf, ist durch diese meist der Eingabekanal (vorzugsweise Kanal 1) vorgegeben.

## 6.5. Programmvergleich

Mit Hilfe des Vergleichsprogrammes kann überprüft werden, ob die abgespeicherten Werte mit dem Inhalt des Lochstreifens übereinstimmen. Unmittelbar nach Programmeingabe können damit insbesondere Lesefehler der Lochbandabtaster erkannt werden, da eine völlig gleichartige Wiederholung von Fehlern sehr unwahrscheinlich ist. Empfehlenswert ist es außerdem, die Programmeingabe über Leser 1 und das Vergleichen über Leser 2 vorzunehmen. Bei Abweichungen der Daten mit 1. KZ 0, 3, 4, 8 oder Punkt erfolgt der Ausdruck der absoluten Speicherplatzadresse, des Lochband- und des Speicherplatzinhaltes unabhängig vom 1. KZ stets oktal oder stets dezimal. Alle anderen 1. KZ wirken wie bei Programmeingabe, insbesondere werden auch System-Sofort-Befehle ausgeführt. Deshalb sind derartige Befehle gemeinsam am Ende des Lochstreifens anzuordnen. Enthalten solche Eingabebefehle die Anwahl 1, bleibt der Automat bei abgeschaltetem Leser 1 im Eingabe-Halt stehen. Nach einem Ausdruck ist stets Kanal 2 angewählt. Wurde bereits vor dem Vergleich das Programm abgearbeitet, werden durch das Programm veränderte Zellen (Arbeitszellen) natürlich ebenfalls ausgedruckt, wenn deren Inhalt auf dem Streifen enthalten ist.

D – 5, wenn oktaler Ausdruck gefordert

D – 6, wenn dezimaler Ausdruck gefordert, weiter wie Programmeingabe, aber ohne „H – G – R"

## 6.6. Programmansprung

Ein Systemprogramm wird mit

H – G – R – B – 0 – a..

angesprungen, wenn es mit dem in Zelle a gespeicherten Systembefehl beginnen soll.

## 6.7. Schrittweise Programmabarbeitung

Bei Ansprung mit

H – G – R – B – 0 – a. – w.

erfolgt vor Abarbeitung des in w gespeicherten Systembefehles Ausdruck von Pseudobefehlszähler PZ und Inhalt des Pseudoakkumulators AC. Anschließend kann das Programm durch Eingabe einer neuen Wartezelle w. fortgesetzt werden oder zunächst mit H – G – R – D – 3 bzw. 4 weitere Ausdrucke gefordert werden, dann Neuansprung des Programmes bei PZ erforderlich.

## 6.8. Überlaufanzeigen

Die rote Lampe „FÜ" brennt.

Überlaufanzeige wird verursacht durch:

1. Transport zum Hauptspeicher und
   a) Überschreitung des durch die Wortlänge vorgegebenen Zahlenbereiches
   b) modifizierten Adressen im Bereich des Systems
2. Verschiebungen, wenn der zulässige Wert vor Verschiebung überschritten ist oder durch Linksverschiebungen der zulässige Bereich überschritten wird
3. Eingaben mit Überschreitung der Eingabebereiche
4. Eingabe von Ziffern nach Eingabe des negativen Vorzeichens
5. Erreichung der in I 7 stehenden Abspeicheradresse bei Satzeingabe
6. Multiplikationsresultate außerhalb des Zahlenbereiches
7. Divisionsresultate außerhalb der Grenzen des Normalwortes
8. Eingabe K = 6 oder 7 oder 9 bei Programmeingabe

Bei 1.–6. steht in der Operationsregisteranzeige des Automaten im Operationsteil 000 und im Adreßteil PZ, bei 7. zur besseren Unterscheidung Operation 004 und Adresse 0000 und bei 8. alles Nullen.

Nach Überlauf muß das Programm erneut angesprungen werden.

## 6.9. Korrektur falscher Plätze

### 6.9.1. Befehle und Konstanten (1. KZ 0, 3, 4, 8 oder Punkt)

| | |
|---|---|
| 21 HLA. | Eingabe Hauptleitadresse |
| 2i NLA. | Eingabe erforderliche Nebenleitadresse |
| 20 RA. | Eingabe der Relativadresse des zu ändernden Platzes |
| 1. KZ (2. KZ) x. | Eingabe berichtigtes Wort |

Sind mehrere Plätze zu korrigieren, entfällt die erneute Eingabe der HLA und bereits eingegebener Nebenleitadressen, bei aufeinanderfolgenden Plätzen auch die Eingabe der Relativadresse. Liegen zwischen zu ändernden Plätzen einzelne Werte, die nicht geändert werden sollen, können diese mit KZ „," übergangen werden.

**6.9.2. Änderung von Miniwort, Füllwort oder Doppelwort**

50 $a_1a_2a_3a_4$ W 55. Eingabe des Sofort-System-Befehles „Eingabe Wort nach Platz $a_1a_2a_3a_4$ über Tastatur"

x. Eingabe Dezimalwort

W : 0 – Füllwort, 1 – Miniwort, 3 – Doppelwort

Sind mehrere Plätze zu korrigieren, muß vor jedem Dezimalwort erneut ein Sofort-System-Befehl eingegeben werden. Miniworte können als Normalworte mit KZ 3 eingegeben werden, wenn das zugehörige Füllwort bzw. das auf dem vorhergehenden Platz stehende Doppelwort erst danach eingegeben wird.

**6.10. Bereichsausstanzen**

D – 8 – a. – e. a = Anfangsadresse des zu stanzenden Bereiches

e = Endadresse + 1°

Die Speicherplatzinhalte werden unabhängig von ihrer Interpretation im Programmablauf formal als Oktalzahlen ausgestanzt. Das anfallende Lochband dient der Zwischenspeicherung von Programm und evtl. Datenfeldern zwecks späterer erneuter Eingabe in den Automaten (absoluter Programmstreifen).

Ausgabebild:

21a.40x.x.... x.:/

**6.11. Bereichsausschreiben oktal**

Durch formales Ausschreiben der Speicherplatzinhalte als Oktalzahlen wird die Basis für ein späteres Vergleichen geschaffen. Die Bedeutung von Maschinen- und Systembefehlen ist aus dem Ausdruck erkennbar, die Umrechnung in Dezimalzahlen ist jedoch nicht ohne weiteres möglich.

D – 3 – a. Abbruch durch Setzen Selektor „SP", danach erneute Eingabe a. möglich.

Ausgabebild:

aaaa xxxx xxxx xxx

Tabelle 1 Externer R300-Code für C 8205

| Lochung | Eingabe-verschl. | S | Zeichen KB | Zeichen GB | L | Ausgabe-verschl. |
|---|---|---|---|---|---|---|
| | 0000 | | 0 | # | L | 020X |
| | 0001 | | 1 | < | L | 001X |
| | 0002 | | 2 | " | L | 002X |
| | 0003 | | 3 | = | L | 023X |
| | 0004 | | 4 | [ | L | 004X |
| | 0005 | | 5 | > | L | 025X |
| | 0006 | | 6 | ' | L | 026X |
| | 0007 | | 7 | ≈ | L | 007X |
| | 0010 | | 8 | ~ | L | 010X |
| | 0011 | | 9 | ] | L | 031X |
| | 0012 | S | ZR | | L | 032X |
| | 0013 | | Leserstopp | | | 013X |
| | 0014 | | $ NS | £ | | 034X |
| | 0015 | S | : | ( | L | 015X |
| | 0016 | S | IS | | | 016X |
| | 0017 | | ⇑ | | | 037X |
| | 0020 | | + | = | L | 040X |
| | 0021 | | a | A | L | 061X |
| | 0022 | | b | B | L | 062X |
| | 0023 | | c | C | L | 043X |
| | 0024 | | d | D | L | 064X |
| | 0025 | | e | E | L | 045X |
| | 0026 | | f | F | L | 046X |
| | 0027 | | g | G | L | 067X |
| | 0030 | | h | H | L | 070X |
| | 0031 | | i | I | L | 051X |
| | 0032 | S | (WRZL) | | | 052X |
| | 0033 | S | . | | L | 073X |
| | 0034 | | WS | | | 054X |
| | 0035 | S | SPE | | | 075X |
| | 0036 | S | IB | | | 076X |
| | 0037 | S | GB | | L | 057X |
| | 0040 | S | – | △ | L | 100X |
| | 0041 | | j | J | L | 121X |
| | 0042 | | k | K | L | 122X |
| | 0043 | | l | L | L | 103X |
| | 0044 | | m | M | L | 124X |
| | 0045 | | n | N | L | 105X |
| | 0046 | | o | O | L | 106X |
| | 0047 | | p | P | L | 127X |
| | 0050 | | q | Q | L | 130X |
| | 0051 | | r | R | L | 111X |
| | 0052 | | WR | | | 112X |
| | 0053 | S | SP | | | 133X |
| | 0054 | | Lo 1 | | | 114X |
| | 0055 | S | Lo A | | | 135X |
| | 0056 | | ⇓ | | | 136X |
| | 0057 | S | KB | | L | 117X |
| | 0060 | S | ' | ; | L | 160X |
| | 0061 | | / | ) | L | 141X |
| | 0062 | | s | S | L | 142X |
| | 0063 | | t | T | L | 163X |
| | 0064 | | u | U | L | 144X |
| | 0065 | | v | V | L | 165X |
| | 0066 | | w | W | L | 166X |
| | 0067 | | x | X | L | 147X |
| | 0070 | | y | Y | L | 150X |
| | 0071 | | z | Z | L | 171X |
| | 0072 | | ZL | | | 172X |
| | 0073 | S | , | ? | L | 153X |
| | 0074 | | % | * | L | 174X |
| | 0075 | S | Tab / WM | | L | 155X |
| | 0076 | | Lo 2 | | | 156X |
| | 0077 | S | | | | 177X |
| | 0032 | S | WRZLISm | | L | 272X |
| | 0072 | S | ⊙ Bm | ⊙ | | 352X |
| | – | – | Irrung | | L | 377X |

S Zeichen stellt Eingabehaltbefehl in Sprung um
L Zeichen kann mit der SE 5L C 8008 gelocht werden

Tabelle 2

| | | |
|---|---|---|
| 0 | = | 00 |
| 1 | < | 01 |
| 2 | ‖ | 02 |
| 3 | ≈ | 03 |
| 4 | [ | 04 |
| 5 | > | 05 |
| 6 | ' | 06 |
| 7 | ≋ | 07 |
| 8 | ~ | 10 |
| 9 | ] | 11 |
| : | ( | 15 |
| + | ǂ | 20 |
| . | ! | 33 |
| — | △ | 40 |
| , | ; | 60 |
| / | ) | 61 |
| % | * | 74 |
| , | ? | 73 |
| a | A | 21 |
| b | B | 22 |
| c | C | 23 |
| d | D | 24 |
| e | E | 25 |
| f | F | 26 |
| g | G | 27 |
| h | H | 30 |
| i | I | 31 |
| j | J | 41 |
| k | K | 42 |
| l | L | 43 |
| m | M | 44 |
| n | N | 45 |
| o | O | 46 |
| p | P | 47 |
| q | Q | 50 |
| r | R | 51 |
| s | S | 62 |
| t | T | 63 |
| u | U | 64 |
| v | V | 65 |
| w | W | 66 |
| x | X | 67 |
| y | Y | 70 |
| z | Z | 71 |
| | ZR | 12 |
| | Tab | 75 |
| | WRZL | 32 |
| | WR | 52 |
| | ZL | 72 |
| | GB | 37 |
| | KB | 57 |

# Systembefehle FIPS v

Tabelle 3

### I. Standardbefehle

| V | S | M | I | Adresse | W | T | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0 | Leerbefehl | |
| | | | | | | | 1 | Addition | |
| | | | | | | | 2 | Subtraktion | |
| | | | | | | | 3 | Multiplikation | |
| | | | | | | | 4 | Division | |
| 0 | AW | | | | | $T^+$ | | | Wort |
| 1 | | | | | 2 | 4 | 5 | Eingabe | Satz 1 |
| 3 | | | | | | | | | Satz 3 |
| 0 | | | | | W | | | | Wort |
| 1 | | | | | 2 | D | 6 | Ausgabe | Satz 1 |
| 3 | | | | | | | | | Satz 3 |
| 2 | | | | | 0 | 0 | | | Text |
| H | 0 | | | | A | B | 7 | Sprung | |

### II. Summenbefehle

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 0 | $M^+$ | I | Adresse | 2 | 4 | 1 | Summenbefehl | Addition |
| | | | | | | | 2 | | Subtraktion |

### III. Sonderbefehle (Auswahl)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | M | I | Markierung | 4 | 2 | 0 | Stopp | unbedingt |
| | | | | | | 6 | | | bedingt |
| | | | | Adresse | 5 | 4 | 0 | Löschen | Zelle |
| | | | | | 7 | | | | Spur |
| | | | | | 4 | 0 | 7 | Sprung i. Masch.-Pr. | |
| | | | | | 4 | 2 | 7 | UP-Spr. i. Masch.-Pr. | |
| | | | | Schl.-Nr. | 4 | 3 | 6 | Ausgabe Zeichen | |
| 2 | $I^+$ | | | Adresse | 5 | 1 | 1 | $\langle I^+ \rangle := \langle a \rangle$ | |
| | | | | | 5 | 4 | 0 | $\langle a \rangle := \langle I^+ \rangle$ | |
| | | | | | 7 | 1 | 1 | $\langle I^+ \rangle :=$ Summe Spur | |
| | | | | | | | 0 | $\langle I^+ \rangle :=$ Summe Spur $+ \langle I^+ \rangle$ | |

A Sprungart

| | |
|---|---|
| 0 | Sprung, wenn Bedingung B erfüllt |
| 1 | wie 0 und I 7 : = PZ + 1° |
| 2 | Sprung, wenn Bedingung B nicht erfüllt |
| 3 | wie 2 und I 7 : = PZ + 1° |

AW Anwahl bzw. SW-Wagenoperation vor Eingabe und Ausgabe

| | | | |
|---|---|---|---|
| 0 | AW Tastatur | 4 | Ausgabe „Punkt" |
| 1 | AW Leser 1 u. Stanzer | 5 | verboten |
| 2 | AW Leser 2 u. Schreibw. | 6 | Ausgabe „Tab" |
| 3 | wirkungslos | 7 | Ausgabe „WRZL" |

B Sprungbedingung

| | | | |
|---|---|---|---|
| 0 | unbedingt | 4 | Selektor „SP" gesetzt |
| 1 | AC ≧ 0 | 5 | ⟨I⟩ = 0 |
| 2 | AC = 0 oder 0– | 6 | ⟨I⟩ = 16 |
| 3 | AC ≦ 0 | 7 | ⟨I⟩ ≦ 16 |

D Nr. des Druckregisters, das die Druckparameter enthält:

| V | S | O | O | $Z_1$ | $Z_2$ | O | N | O | O | SZ |
|---|---|---|---|---|---|---|---|---|---|---|

V Verschiebeanweisung (nur vor Ausgabe Wort wirksam)
S Stellenzahl der Verschiebung V
Z Zahl der Stellen vor dem Komma (für Z > 9 oktal codieren)
N Zahl der Stellen nach dem Komma (0–8)
SZ Schlußzeichen
SZ = 5: Ausgabe „ZR", wenn AC ≧ 0
sonst SZ wie AW

H Halt vor Sprungausführung (bei nicht ausgeführtem Sprung wirkungslos)

| | |
|---|---|
| 0 | kein Stopp |
| 2 | Stopp, wenn Selektor „BS" gesetzt |
| 3 | Stopp |

I Nr. des Indexregisters für Befehlsmodifikation und Sprungbedingung B

I+ Nr. des Indexregisters bei Abarbeitung von Sonderbefehlen

M Befehlsmodifikation

| | | | |
|---|---|---|---|
| 0 | keine Modifikation | 4 | Registertransport |
| 1 | Substitution | 5 | 1, dann 4 |
| 2 | Änderung | 6 | 2, dann 4 |
| 3 | 1, dann 2 | 7 | 1, dann 2, dann 4 |

M+ Befehlsmodifikation bei Summenbefehlen

| | |
|---|---|
| 0 | keine Modifikation |
| 2 | Adressenänderung |

S Stellenzahl bei dezimaler Verschiebung

T Transportanweisung (entspr. W)

| | | | |
|---|---|---|---|
| 0 | kein Transport | 4 | ⟨a⟩ : = AC nach Op und V |
| 1 | AC : = ⟨a⟩ vor Op und V | 5 | 1 und 4 |
| 2 | R : = ⟨a⟩ vor Op und V | 6 | 2 und 4 |
| 3 | 1 und 2 | 7 | 1 und 2 und 4 |

| | |
|---|---|
| T+ | Transportanweisung bei Eingabe Wort |
| 1 | Eingabe Wort nach AC |
| 3 | 1, dann R : = AC |
| 5 | 1, dann ⟨a⟩ : = AC entspr. W |
| 7 | 1, dann R : = AC, dann ⟨a⟩ : = AC entspr. W |
| V | Verschiebeanweisung um S Dezimalstellen |
| 0 | Rechtsverschiebung ohne Rundung |
| 1 | Rechtsverschiebung mit kaufmännischer Rundung |
| 2 | Linksverschiebung |
| 3 | Linksverschiebung und Betragsbildung |
| W | Wortlänge bei Transporten von und zu dem Hauptspeicher |
| 0 | Füllwort $0 \leq x < 4096$ |
| 1 | Miniwort 6stellig |
| 2 | Normalwort 9/10stellig |
| 3 | Doppelwort 13stellig |
| | 1 Füllwort + 1 Miniwort / 1 Zelle |
| | 1 Doppelwort + 1 Miniwort / 2 benachb. Zellen |

Tabelle 4

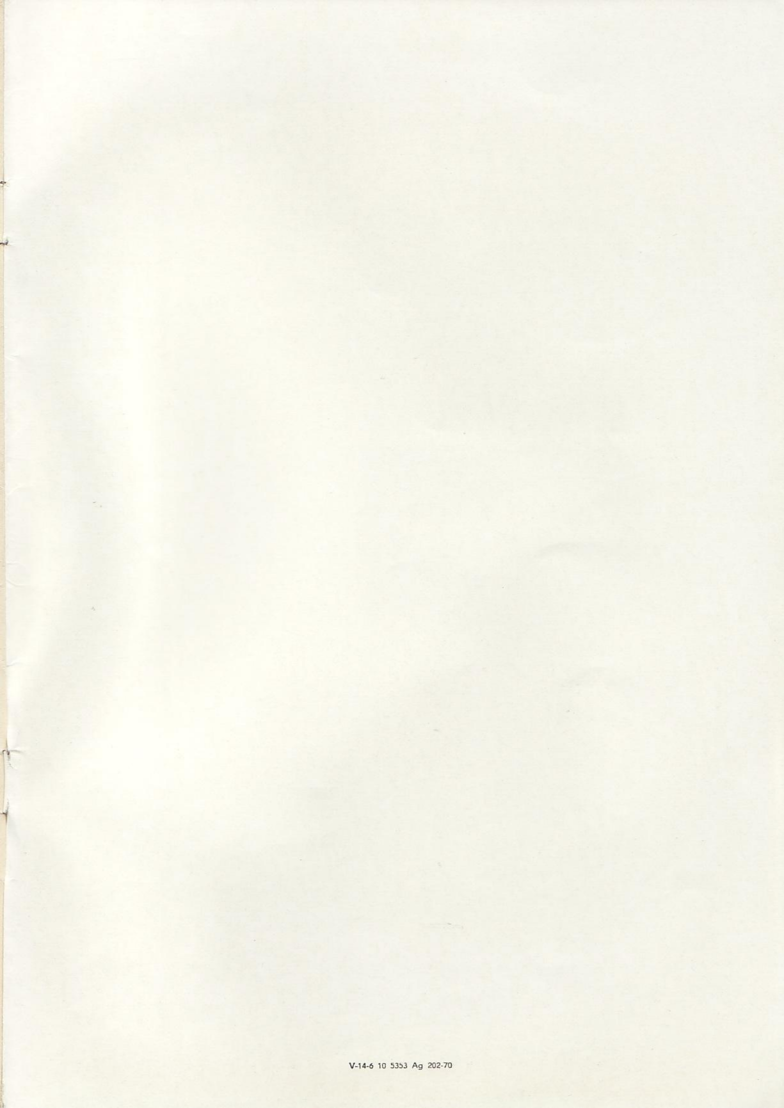
| | Programm-Nr. 01<br>Beispiel Bestandsrechnung | Blatt 1<br>von 1 |
|---|---|---|

Bemerkungen:

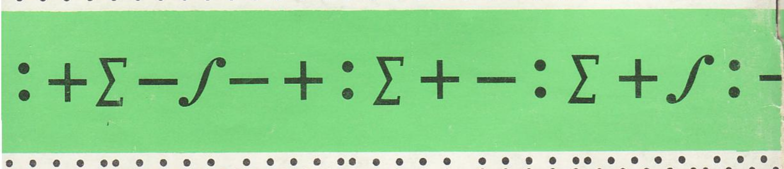
2. LA Anfang Speicherbereich Mengen und Werte nach Gruppen saldiert
3. LA Gesamtsumme

| Nr. | | | | | | | | | | | | | R | AC | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 02 | | | 4 | 1 | 0 | 0 | 0 | 0 | 7 | 4 | 0 | | | I1 : = A°, Lö 1. Spur |
| 01 | 02 | | | 4 | 6 | 1 | 7 | 4 | 0 | 0 | 0 | 0 | | | I6 : = A + 1740° |
| 02 | 00 | | | 6 | 1 | 0 | 0 | 4 | 0 | 7 | 4 | 0 | | | I1 : = I; + 40°, Lö. Spur i |
| 03 | 01 | | | | 1 | 0 | 0 | 0 | 2 | 2 | 6 | 7 | | | Sprung bei I1 ≠ I6 |
| 04 | 02 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 5 | | | Satzeing. 3 über Leser 1 |
| 05 | 02 | 2 | 2 | 4 | 1 | 0 | 0 | 0 | 0 | 7 | 1 | 1 | | | I1 : = A°. I2 : = Summe |
| 06 | 00 | 2 | 2 | 6 | 1 | 0 | 0 | 4 | 0 | 7 | 1 | 0 | | | I1 : = A + 40°, Summier. |
| 07 | 01 | | | | 1 | 0 | 0 | 0 | 6 | 2 | 6 | 7 | | | Sprung bei I1 ≠ I6 |
| 10 | 00 | | 1 | 0 | 0 | 7 | 7 | 6 | 6 | 2 | 5 | 5 | | | I6 : = Kontrollsumme |
| 11 | 01 | 3 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 6 | 7 | | | Stopp bei I2 ≠ I6 |
| 12 | 00 | | | 4 | 6 | 0 | 7 | 7 | 5 | 0 | 0 | 0 | | | I6 : = 775° (dez. 509) |
| 13 | 00 | | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 5 | 5 | | | I1 : = Nr. v. Leser 2 |
| 14 | 01 | | | | | | | 2 | 4 | 0 | 0 | 7 | | | |
| 15 | 02 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 6 | | | Ausgabe Satz 3 auf LB |
| 16 | 02 | 2 | 2 | 4 | 1 | 0 | 0 | 0 | 0 | 7 | 1 | 1 | | | I1 : = A°, summieren |
| 17 | 02 | 0 | 0 | 4 | 6 | 1 | 7 | 4 | 0 | 0 | 0 | 0 | | | I6 : = A + 1740° |
| 20 | 00 | 2 | 2 | 6 | 1 | 0 | 0 | 4 | 0 | 7 | 1 | 0 | | | I1 : = I1 + 40°, summ. |
| 21 | 01 | | | | 1 | 0 | 0 | 2 | 0 | 2 | 6 | 7 | | | Sprung bei I1 ≠ I6 |
| 22 | 00 | 0 | 1 | 0 | 0 | 7 | 7 | 6 | 2 | 2 | 0 | 6 | | | Ausg. Kontrollsumme |
| 23 | 01 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | | | unbed. Stopp |
| 24 | 01 | | | | 1 | 0 | 0 | 2 | 7 | 0 | 7 | 7 | | | Sprung bei Nr. ≦ 509 |
| 25 | 00 | | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 6 | | | WRZL, Ausg.-Nr. |
| 26 | 00 | | | 4 | 1 | 0 | 7 | 7 | 6 | 4 | 6 | 0 | | | bed. Stopp, Nr. : = 510 |
| 27 | 00 | | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 5 | M | M | |
| 30 | 02 | 4 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 4 | 1 | M | M | M (Nr.) : = M (Nr.) + M |
| 31 | 03 | | | | | | | | | 3 | 5 | 1 | M | M ges. | |
| 32 | 00 | | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 5 | M | P | |
| 33 | 00 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | M | W | kaufm. Rundung |
| 34 | 02 | 4 | 0 | 2 | 1 | 1 | 0 | 0 | 0 | 2 | 4 | 1 | M | W | W (Nr.) : = W (Nr.) + W |
| 35 | 03 | | | | | | | | 2 | 3 | 6 | 1 | W ges. | W ges. | |
| 36 | 01 | | | | | | | 1 | 3 | 0 | 0 | 7 | | | z. Einlesen Nr. |
| 37 | | | | | | | | | | | | | | | |

| Bearbeiter: | Geprüft: | Stand: |
|---|---|---|



V-14-6 10 5353 Ag 202-70



**VEB KOMBINAT ZENTRONIK**
**Rechenelektronik Meiningen, Sitz Zella-Mehlis**
DDR - 606 Zella-Mehlis
Deutsche Demokratische Republik

Exporteur:
**BÜROMASCHINEN-EXPORT GmbH.**
DDR - 108 Berlin 8
Friedrichstraße 61